Pairwise Alignment and Database Searching

Henrik Nielsen
Protein Post-Translational Modification &
Molecular Evolution Groups
Center for Biological Sequence Analysis
If two sequences look similar, the explanation may be:

• Homology
  (common descent)

• Convergent evolution
  (common function \(\rightarrow\) common selective pressure)

• Chance!
Sequences are related

- Darwin: all organisms are related through descent with modification
- => Sequences are related through descent with modification
- => Similar molecules have similar functions in different organisms

Phylogenetic tree based on ribosomal RNA:
three domains of life

**ARCHAEA**
- Methanococcus
- Methanobacterium
- Methanosarcina

**BACTERIA**
- Escherichia coli
- Agrobacterium
- Chlamydomonas
- Clostridium
- Chlorella
- Bacillus
- Synechococcus
- Thermus
- Thermotoga
- Thermus thermophilus

**EUCARYA**
- Yeast
- Homo
- Saccharomyces
- Paramaecium
- Polypora
- Diatoma
- Euglena
- Volvox
- Chlamydomonas
- Nematodes
- Forcigera
Sequences are related, II

Phylogenetic tree of globin-type proteins found in humans
Why compare sequences?

- Determination of evolutionary relationships
- Prediction of protein function and structure (database searches).

Protein 1: binds oxygen

Sequence similarity

Protein 2: binds oxygen?
Dotplots: visual sequence comparison

1. Place two sequences along axes of plot

3. Place dot at grid points where two sequences have identical residues

5. Diagonals correspond to conserved regions
Pairwise alignments

43.2% identity; Global alignment score: 374

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>V-LSPADKTVKAAWGKVGAHAGEYGAELSEALERMFLSFPTTKYFPHF-DLS----HGSA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVYVPTQRFFESFGDLSTPDADVGMNP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAKLRVDPVNFKLLSRLTLVTLAHL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>KVKAHGKKVLGAFSDLHDLNLKGFATLSELHCDKLHVDPENFRLGGNVLVCVLHAF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>PAEFTPAPHAVSDLKFLASVSTVLTSKYR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>GKEFTPFVQAAYQKVAVANALAHKYH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pairwise alignment

100.000% identity in 3 aa overlap

SPA
:::
SPA

Percent identity is not a good measure of alignment quality
Pairwise alignments: alignment score

43.2% identity;  Global alignment score: 374

```
alpha  V-LSPADKTNVKAARGKVGAGAGEYGAELERMFLSFPTTKTYFPHF-DLS-----HGSA
beta   VHLTPEEKSAVTALWGVK--NVDEVGGEALGRLLVYPTQRFFESFGDLSTPDAVGMNP
    10          20        30        40        50
```

```
alpha  QVKGHGKKVADALTNAVHVDDMPNALSALSDHLAHKLRLVDPVNFKLLSHCLLVTLLAAHL
beta   KVKAHGKKVGLFGSDGLAHLDNLKGTFTATLSHELHCDKHLVDPEFRLLLGNNVLCVLAHF
    60        70        80        90       100       110
```

```
alpha  PAEFTPVAHSALDKFLASVSTVLTSKYR
beta   GKEFTPVQAAYQKVAGVANALAHKYH
   120       130       140
```
Alignment scores: match vs. mismatch

Simple scoring scheme (too simple in fact…):

Matching amino acids: 5
Mismatch: 0

Scoring example:

K A W S A D V
: : : : :
K D W S A E V

5+0+5+5+5+0+5 = 25
Pairwise alignments: conservative substitutions

43.2% identity;  Global alignment score: 374

10  20  30  40  50
alpha  V-LSPADKTNVKAAWGKVGGAHAGEYGAEEALERMFLSFPTTKTYFPFH-DLS----HGSA
       : :::: ··· ·::: ·::: :·::: ::·: :·: :·: :·: ·::
beta  VHLTPEEEKSAVTALWGKV--NVDEVGGEALGRLLVVPWTQRFFESFDIISTPDAVMGNP
       10  20  30  40  50

60  70  80  90  100  110
alpha  QVKGHGKKVADALTNAVAHVDMPNALSALSDLHAHKLRLVDPVNFKLLSHCLLVTLAAHL
       : :::: :··········· ···································
beta  KVKAHGKVLGAFSGDLALDNLGTFATLSELHCDKLHVDPLPFRLLGNVLVCVLAHHF
       60  70  80  90  100  110

120 130 140
alpha  PAEFTPAVHASLKDFSVSTATLTSKYR
       :··················································
beta  GKEFTPFPVQAAYQKVAGVANALAHKYH
       120 130 140
Amino acid properties

Serine (S) and Threonine (T) have similar physicochemical properties

Aspartic acid (D) and Glutamic acid (E) have similar properties

Substitution of S/T or E/D occurs relatively often during evolution

Substitution of S/T or E/D should result in scores that are only moderately lower than identities
**Protein substitution matrices**

|    | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
| A  | 5 | -2 | -1 | -2 | -1 | 1 | 1 | 0 | -1 | -1 | -2 | -3 | -4 | -3 | -4 | -3 | -4 | -3 | -4 | 5 |
| R  | -2 | 7 | -2 | -2 | 2 | 8 | -3 | -3 | -3 | -3 | -2 | -3 | -4 | -3 | -4 | -3 | -4 | -3 | -4 | 5 |
| N  | -1 | -1 | 7 | -2 | -4 | -2 | -4 | 13 | -1 | 1 | 0 | 0 | -3 | 7 | -1 | 0 | 0 | -1 | -1 | -1 | 5 |
| D  | -2 | -2 | 2 | 8 | -2 | -2 | -3 | 2 | 8 | -2 | -3 | 2 | 6 | -3 | -2 | -3 | -4 | -3 | -4 | 2 | 5 |
| C  | -1 | 1 | 0 | 0 | -3 | 1 | 0 | -2 | 10 | 1 | 0 | 0 | -3 | -3 | 6 | 1 | 0 | -2 | -2 | 10 |
| Q  | -1 | 0 | 0 | 2 | -3 | 2 | 6 | 0 | -3 | 0 | -1 | -3 | -2 | -3 | 8 | 0 | -3 | 0 | -1 | -3 | 8 |
| E  | -1 | 0 | 0 | -1 | -3 | -2 | -3 | 8 | 1 | 0 | 0 | -2 | 10 | 0 | -3 | -3 | -4 | -3 | -4 | 2 | 5 |
| G  | 0 | -3 | 0 | -1 | -3 | -2 | -3 | 8 | 0 | -3 | 0 | -1 | -3 | -2 | -3 | 8 | 0 | -3 | 0 | -1 | -3 | 8 |
| H  | -2 | 0 | 1 | -1 | -3 | 1 | 0 | -2 | 10 | 1 | 0 | 0 | -3 | -3 | 6 | 1 | 0 | -2 | -2 | 10 |
| I  | -1 | -4 | -3 | -4 | -2 | -3 | -4 | -4 | -4 | -2 | -4 | -2 | -3 | -4 | -4 | -3 | -4 | -4 | -4 | 5 |
| L  | -2 | -3 | -4 | -4 | -2 | -2 | -3 | -4 | -3 | -2 | -3 | -4 | -3 | -2 | -3 | -4 | -3 | -2 | -3 | 5 |
| K  | -1 | 3 | 0 | -1 | -3 | 2 | 1 | -2 | 0 | -3 | -3 | 6 | -1 | 3 | 0 | -1 | -3 | -3 | 6 | 5 |
| M  | -1 | -2 | -2 | -4 | -2 | 0 | -2 | -3 | -1 | 2 | 3 | -2 | 7 | -1 | -2 | -2 | -4 | -1 | -3 | -4 | 10 |
| F  | -3 | -3 | -4 | -5 | -2 | -4 | -3 | -4 | -1 | 0 | 1 | -4 | 0 | 8 | -3 | -3 | -4 | -5 | -2 | -4 | -3 | 15 |
| P  | -1 | -3 | -2 | -1 | -4 | -1 | -1 | -2 | -2 | -3 | -4 | -1 | -3 | -4 | 10 | -1 | -3 | -2 | -1 | -1 | -3 | -2 | 10 |
| S  | 1 | -1 | 1 | 0 | -1 | 0 | -1 | -3 | -3 | 0 | -2 | -3 | -1 | 5 | -1 | 1 | 0 | -1 | -3 | -3 | -1 | 5 |
| T  | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | -1 | -1 | -2 | -1 | 2 | 5 | 0 | -1 | -1 | -1 | -2 | 5 |
| W  | -3 | -3 | -4 | -5 | -5 | -1 | -3 | -3 | -3 | -3 | -2 | -3 | -1 | 1 | -4 | -4 | -3 | 15 | -1 | -3 | -2 | -2 | 2 | 8 |
| Y  | -2 | -1 | -2 | -3 | -3 | -1 | -2 | -3 | 2 | -1 | -1 | -2 | 0 | 4 | -3 | -2 | -2 | 2 | 8 | -2 | -1 | -2 | 2 | 8 |
| V  | 0 | -3 | -3 | -4 | -1 | -3 | -3 | -4 | -4 | -4 | 4 | 1 | -3 | 1 | -1 | -3 | -2 | 0 | -3 | -1 | 5 | -1 | -3 | -1 | 5 |

**BLOSUM50 matrix:**

- Positive scores on diagonal (identities)
- Similar residues get higher (positive) scores
- Dissimilar residues get smaller (negative) scores
Pairwise alignments: insertions/deletions

43.2% identity;  
Global alignment score: 374

```
10  20  30  40  50
alpha V-LSPADKTNVKAAWGKVGAHAGEYGAEEALERMFLSFPFTKTYFPHF-DLS-----HGSA
beta  VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAV

10  20  30  40  50
60  70  80  90 100 110
alpha QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAKLRVDPVNFKLLLLHCLLVTLAAHL
beta  KVKAHGKKVLGAFSDGLAHLDNLKGTATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF

60  70  80  90 100 110
120 130 140
alpha PAEFTPAVHASLDKFLASVSTVLTSKYR
beta  GKEFTPFVQAAYQKVAGVANALAHKYH
```

120  130  140
Alignment scores: insertions/deletions

Affine gap penalties:

Multiple insertions/deletions may be one evolutionary event =>
Separate penalties for gap opening and gap elongation
Compute 4 alignment scores: two different alignments using two different alignment matrices (and the same gap penalty system)

Score 1: Alignment 1 + BLOSUM-50 matrix + gaps
Score 2: Alignment 1 + BLOSUM-Trp matrix + gaps
Score 3: Alignment 2 + BLOSUM-50 matrix + gaps
Score 4: Alignment 2 + BLOSUM-Trp matrix + gaps
### Handout: summary of results

<table>
<thead>
<tr>
<th></th>
<th>Alignment 1</th>
<th>Alignment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BLOSUM-50</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BLOSUM-Trp</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Protein substitution matrices

|   | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
| A | 5 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| R | -2 | 7 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| N | -1 | -1 | 7 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| D | -2 | -2 |   | 2 | 8 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| C | -1 | -4 | -2 | -4 | 13 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Q | -1 | 1 | 0 | 0 | -3 | 7 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| E | -1 | 0 | 0 | 2 | -3 | 2 | 6 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| G | 0 | -3 | 0 | -1 | -3 | -2 | -3 | 8 |   |   |   |   |   |   |   |   |   |   |   |   |   |
| H | -2 | 0 | 1 | -1 | -3 | 1 | 0 | -2 | 10 |   |   |   |   |   |   |   |   |   |   |   |   |
| I | -1 | -4 | -3 | -4 | -2 | -3 | -4 | -4 | -4 | 5 |   |   |   |   |   |   |   |   |   |   |   |
| L | -2 | -3 | -4 | -4 | -2 | -2 | -3 | -4 | -3 | 2 | 5 |   |   |   |   |   |   |   |   |   |   |   |
| K | -1 | 3 | 0 | -1 | -3 | 2 | 1 | -2 | 0 | -3 | -3 | 6 |   |   |   |   |   |   |   |   |   |   |
| M | -1 | -2 | -2 | -4 | -2 | 0 | -2 | -3 | -1 | 2 | 3 | -2 | 7 |   |   |   |   |   |   |   |   |   |   |
| F | -3 | -3 | -4 | -5 | -2 | -4 | -3 | -4 | -1 | 0 | 1 | -4 | 0 | 8 |   |   |   |   |   |   |   |   |   |   |
| P | -1 | -3 | -2 | -1 | -4 | -1 | -1 | -2 | -2 | -3 | -4 | -1 | -3 | -4 | 10 |   |   |   |   |   |   |   |   |   |   |
| S | 1 | -1 | 1 | 0 | -1 | 0 | -1 | 0 | -1 | -3 | -3 | 0 | -2 | -3 | -1 | 5 |   |   |   |   |   |   |   |   |   |   |
| T | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | -1 | -1 | -2 | -1 | 2 | 5 |   |   |   |   |   |   |   |   |
| W | -3 | -3 | -4 | -5 | -1 | -3 | -3 | -3 | -2 | -3 | -1 | 1 | -4 | -4 | -3 | 15 |   |   |   |   |   |   |   |   |   |
| Y | -2 | -1 | -2 | -3 | -3 | -1 | -2 | -3 | 2 | -1 | -1 | -2 | 0 | 4 | -3 | -2 | -2 | 2 | 8 |   |   |   |   |   |
| V | 0 | -3 | -3 | -4 | -1 | -3 | -3 | -4 | -4 | 4 | 1 | -3 | 1 | -1 | -3 | -2 | 0 | -3 | -1 | 5 |   |   |   |   |

BLOSUM50 matrix:

- Positive scores on diagonal (identities)
- Similar residues get higher (positive) scores
- Dissimilar residues get smaller (negative) scores
Protein substitution matrices: different types

- **Identity matrix**
  (match vs. mismatch)

- **Genetic code matrix**
  (how similar are the codons?)

- **Chemical properties matrix**
  (use knowledge of physicochemical properties to design matrix)

- **Empirical matrices**
  (based on observed pair-frequencies in hand-made alignments)
  - PAM series
  - BLOSUM series
  - Gonnet
Empirical matrices: the problem

How to use existing alignments, when the scoring system is not defined yet?

- The PAM solution:
  - Use global alignments of very closely related sequences, and then extrapolate to larger distances by matrix multiplication

- The BLOSUM solution:
  - Use only gap-free alignments of highly conserved regions

- The Gonnet solution:
  - Use an evolutionary model
Estimation of a BLOSUM matrix

- The BLOCKS database contains local multiple gap-free alignments of proteins.
- All pairs of amino acids in each column of each BLOCK are compared, and the observed pair frequencies are noted (e.g., A aligned with A makes up 1.5% of all pairs; A aligned with C makes up 0.01% of all pairs, etc.)
- Expected pair frequencies are computed from single amino acid frequencies. (e.g., \( f_{A,C} = f_A \times f_C = 7\% \times 3\% = 0.21\% \)).
- For each amino acid pair the substitution scores are essentially computed as:

\[
S_{A,C} = \log \frac{\text{Pair-freq(obs)}}{\text{Pair-freq(expected)}}
\]

\[
S_{A,C} = \log \frac{0.01\%}{0.21\%} = -1.3
\]
Estimation of a BLOSUM a matrix, cont'd

- For each alignment in the BLOCKS database, the sequences are grouped into clusters with at least 50% identical residues (for BLOSUM 50)

- Only pairs of sequences *between* clusters are compared while calculating amino acid pair frequencies

- A lower BLOSUM number means a lower average % identity, and therefore a larger evolutionary distance
Pairwise alignment

Optimal alignment:

alignment having the highest possible score given a substitution matrix and a set of gap penalties
Pairwise alignment: the problem

The number of possible pairwise alignments increases explosively with the length of the sequences:

Two protein sequences of length 100 amino acids can be aligned in approximately $10^{60}$ different ways.

Time needed to test all possibilities is same order of magnitude as the entire lifetime of the universe.
Pairwise alignment: the solution

"Dynamic programming"
(the Needleman-Wunsch algorithm)
Alignment depicted as path in matrix

TCGCA
TC-CA

TCGCA
T-CCA
Alignment depicted as path in matrix

Meaning of point in matrix: all residues up to this point have been aligned (but there are many different possible paths).

Position labeled “x”: TC aligned with TC

```
T C G C A
```

```
T
C
C
A
```

```
--TC
TC--
```

```
-TC
T-C
```

```
TC
TC
```
Dynamic programming: computation of scores

Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.
Dynamic programming: computation of scores

Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

\[
\text{score}(x,y) = \max \begin{cases} 
\text{score}(x,y-1) - \text{gap-penalty} 
\end{cases}
\]
Dynamic programming: computation of scores

Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

$$\text{score}(x,y) = \max \begin{cases} 
\text{score}(x,y-1) - \text{gap-penalty} \\
\text{score}(x-1,y-1) + \text{substitution-score}(x,y)
\end{cases}$$
Dynamic programming: computation of scores

Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

score(x,y) = max

\[
\begin{align*}
\text{score}(x,y-1) - \text{gap-penalty} \\
\text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\
\text{score}(x-1,y) - \text{gap-penalty}
\end{align*}
\]
Any given point in matrix can only be reached from three possible positions (you cannot “align backwards”).

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

Each new score is found by choosing the maximum of three possibilities. For each square in matrix: keep track of where best score came from.

Fill in scores one row at a time, starting in upper left corner of matrix, ending in lower right corner.

\[
\text{score}(x,y) = \max \begin{cases} 
\text{score}(x,y-1) - \text{gap-penalty} \\
\text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\
\text{score}(x-1,y) - \text{gap-penalty} 
\end{cases}
\]
Dynamic programming: example

\[
a[i,j] = \max \begin{cases} 
a[i,j-1] - 2 \\
a[i-1,j-1] + p(i,j) \\
a[i-1,j] - 2 \\
\end{cases}
\]
Dynamic programming: example

\[
a[i,j] = \max \begin{cases} 
a[i,j-1] - 2 \\
a[i-1,j-1] + p(i,j) \\
a[i-1,j] - 2 
\end{cases}
\]
Dynamic programming: example

\[
a[i,j] = \max \begin{cases} 
    a[i,j-1] - 2 \\
    a[i-1,j-1] + p(i,j) \\
    a[i-1,j] - 2 
\end{cases}
\]
Dynamic programming: example

\[
a[i,j] = \max \begin{cases} 
a[i,j-1] - 2 \\
a[i-1,j-1] + p(i,j) \\
a[i-1,j] - 2
\end{cases}
\]
Dynamic programming: example
Dynamic programming: example

\[
\begin{align*}
0 & \quad -2 & \quad -4 & \quad -6 & \quad -8 & \quad -10 \\
-2 & \quad 1 & \quad -1 & \quad -3 & \quad -5 & \quad -7 \\
-4 & \quad -1 & \quad 2 & \quad 0 & \quad -2 & \quad -4 \\
-6 & \quad -3 & \quad 0 & \quad 1 & \quad 1 & \quad -1 \\
-8 & \quad -5 & \quad -2 & \quad -1 & \quad 0 & \quad 2
\end{align*}
\]

\[
\begin{align*}
TCGCA \\
\begin{array}{cccc}
: & : & : & : \\
T & C & \underline{G} & CA
\end{array}
\end{align*}
\]

\[
1+1-2+1+1 = 2
\]
Global versus local alignments

Global alignment: align full length of both sequences. (The “Needleman-Wunsch” algorithm).

Local alignment: find best partial alignment of two sequences (the “Smith-Waterman” algorithm).
Local alignment overview

- The recursive formula is changed by adding a fourth possibility: zero. This means local alignment scores are never negative.

\[
\text{score}(x,y) = \max \begin{cases} 
\text{score}(x,y-1) - \text{gap-penalty} \\
\text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\
\text{score}(x-1,y) - \text{gap-penalty} \\
0
\end{cases}
\]

- Trace-back is started at the highest value rather than in lower right corner.
- Trace-back is stopped as soon as a zero is encountered.
## Local alignment: example

<table>
<thead>
<tr>
<th></th>
<th>H</th>
<th>E</th>
<th>A</th>
<th>G</th>
<th>A</th>
<th>W</th>
<th>G</th>
<th>H</th>
<th>E</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>20</td>
<td>12</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>H</td>
<td>0</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>18</td>
<td>22</td>
<td>14</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>18</td>
<td>28</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>21</td>
<td>13</td>
<td>5</td>
<td>0</td>
<td>4</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>13</td>
<td>18</td>
<td>12</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

**AWGHE**

**AW-HE**
Substitution matrices and sequence similarity

• Substitution matrices come as series of matrices calculated for different
degrees of sequence similarity (different evolutionary distances).

• ”Hard” matrices are designed for similar sequences
  – Hard matrices are designated by high numbers in the BLOSUM series (e.g., BLOSUM80)
  – Hard matrices yield short, highly conserved alignments

• ”Soft” matrices are designed for less similar sequences
  – Soft matrices have low BLOSUM values (45)
  – Soft matrices yield longer, less well conserved alignments
“Optimal alignment” means “having the highest possible score, given substitution matrix and set of gap penalties”.

This is NOT necessarily the biologically most meaningful alignment.

Specifically, the underlying assumptions are often wrong: substitutions are not equally frequent at all positions, affine gap penalties do not model insertion/deletion well, etc.

Pairwise alignment programs always produce an alignment - even when it does not make sense to align sequences.
Database searching

Using pairwise alignments to search databases for similar sequences

Query sequence

Database

In principle: compute pairwise alignments between your query sequence and every database sequence
Database searching

• Most common use of pairwise sequence alignments: to search databases for related sequences.

• Most often, *local* alignment is used for database searching:
  
  you want to know whether *something* matches, not whether *everything* matches.

• Often, full Smith-Waterman is too time-consuming for searching large databases, so heuristic methods are used.
FASTA (Pearson 1995)

Uses heuristics to avoid calculating the full dynamic programming matrix

Speed up searches by an order of magnitude compared to full Smith-Waterman
**Heuristic search algorithms 2**

**BLAST** (Altschul 1990, 1997)

Uses rapid word lookup methods to completely skip most of the database entries.

Extremely fast:
- **One order of magnitude** faster than FASTA
- **Two orders of magnitude** faster than Smith-Waterman

Almost as sensitive as FASTA, but the statistical side of FASTA is still stronger than BLAST.
<table>
<thead>
<tr>
<th>BLAST flavors</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BLASTN</strong></td>
<td><strong>TBLASTN</strong></td>
</tr>
<tr>
<td>Nucleotide query sequence</td>
<td>Protein query sequence</td>
</tr>
<tr>
<td>Nucleotide database</td>
<td>Nucleotide database</td>
</tr>
<tr>
<td><strong>BLASTP</strong></td>
<td></td>
</tr>
<tr>
<td>Protein query sequence</td>
<td></td>
</tr>
<tr>
<td>Protein database</td>
<td></td>
</tr>
<tr>
<td><strong>BLASTX</strong></td>
<td></td>
</tr>
<tr>
<td>Nucleotide query sequence</td>
<td></td>
</tr>
<tr>
<td>Protein database</td>
<td></td>
</tr>
<tr>
<td>Compares all six reading frames with the database</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;On the fly&quot; six frame translation of database</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TBLASTX</strong></td>
<td></td>
</tr>
<tr>
<td>Nucleotide query sequence</td>
<td></td>
</tr>
<tr>
<td>Nucleotide database</td>
<td></td>
</tr>
<tr>
<td>Compares all reading frames of query with all reading frames of the database</td>
<td></td>
</tr>
</tbody>
</table>
Searching on the web: BLAST at NCBI

Very fast computer dedicated to running BLAST searches

Many databases that are always up to date

Nice simple web interface

But you still need knowledge about BLAST to use it properly
When is a database hit significant?

- **Problem:**
  - Even unrelated sequences can be aligned (yielding a low score)
  - How do we know if a database hit is meaningful?
  - When is an alignment score sufficiently high?

- **Solution:**
  - Determine the range of alignment scores you would expect to get for random reasons (i.e., when aligning unrelated sequences).
  - Compare actual scores to the distribution of random scores.
  - Is the real score much higher than you’d expect by chance?
Random alignment scores follow extreme value distributions

Searching a database of unrelated sequences result in scores following an extreme value distribution

The exact shape and location of the distribution depends on the exact nature of the database and the query sequence
Significance of a hit: one possible solution

(1) Align query sequence to all sequences in database, note scores

(2) Fit actual scores to a mixture of two sub-distributions: (a) an extreme value distribution and (b) a normal distribution

(3) Use fitted extreme-value distribution to predict how many random hits to expect for any given score (the “E-value”)

![Graph showing distribution with fitted curves]
Significance of a hit: example

Search against a database of 10,000 sequences.

An extreme-value distribution (blue) is fitted to the distribution of all scores.

It is found that 99.9% of the blue distribution has a score below 112.

This means that when searching a database of 10,000 sequences you’d expect to get 0.1% * 10,000 = 10 hits with a score of 112 or better for random reasons. 10 is the E-value of a hit with score 112. You want E-values well below 1!
Database searching: E-values in BLAST

BLAST uses precomputed extreme value distributions to calculate E-values from alignment scores.

For this reason BLAST only allows certain combinations of substitution matrices and gap penalties.

This also means that the fit is based on a different data set than the one you are working on.

A word of caution: BLAST tends to overestimate the significance of its matches.

E-values from BLAST are fine for identifying sure hits. One should be careful using BLAST’s E-values to judge if a marginal hit can be trusted (e.g., you may want to use E-values of $10^{-4}$ to $10^{-5}$).