Pairwise Alignment

Anders Gorm Pedersen
Molecular Evolution Group
Center for Biological Sequence Analysis
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
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

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

3. 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><strong>beta</strong></td>
<td>VHLTPEEKSAVTALT WGVK--NVDEVGGGEALGRLLVYPWTRFESFGDLSTPDAVMGNP</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td><strong>beta</strong></td>
<td>KVKAHGKKVGLAGSDGLAHLDNKGTATLSELHCDKLHVDPENFRLLGNVLCVLAAHF</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>130</td>
<td>140</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>beta</strong></td>
<td>GKEFTPFPVQAAYQKVAGVANAHKYH</td>
<td>120</td>
<td>130</td>
<td>140</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
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-LSPADKTNVKAAWGKVGAHAGEYGAEEALERMFLSFPTTKTYFPHF-DLS----HGSA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>alpha</td>
<td>QVKGHGKVKVADALTNAVAHAVDDMPNASLSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>KVKAHGKKVLGAFSDGLAHLDNLKGFATLSELHCDKLVHDPENFRLLGNVLVCVLAAHF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>100</td>
</tr>
<tr>
<td>alpha</td>
<td>PAEFTPAVHASLDKFASVSTVLSKYR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>GKEFTPFPVQAAYQKVAGVANALAHKYH</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>130</td>
<td>140</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Alignment scores: match vs. mismatch

Simple scoring scheme (too simple in fact...):

Matching amino acids: 5
Mismatch: 0

Scoring example:

\[
\begin{array}{ccccccc}
K & A & W & S & A & D & V \\
: & : & : & : & : & : & : \\
K & D & W & S & A & E & V \\
\end{array}
\]

\[
5+0+5+5+5+0+5 = 25
\]
Pairwise alignments: conservative substitutions

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-LSPADKTNKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPFH-DLS-----HGSA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>VHLTPEKSAVTLWGKV--NVDEVGGEALGRLVVYPWTQRFFESFGDLSPTDAVMGNP</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>QVKGHGKVKADALTNAHVDDMNALSALSLLAHKLRLVDPVNFKLLSHCLLVTLLAHL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>KVKAHGKKVGLFSDLGHDLNLKGTATLSELHCDKLHVDPMFRLGKVNLVCVLAHHF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>120</th>
<th>130</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>PAEFTPAVHASLDKFLASVSTVLTSKYR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>GKEFTPPPQQAYQKVAGVANALAHKYH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

#### BLOSUM50 matrix:

- Positive scores on diagonal (identities)
- Similar residues get higher (positive) scores
- Dissimilar residues get smaller (negative) scores

|   | A | R | N | D | C | Q | E | G | H | I | L | K | M | F | P | S | T | W | Y | V |
| A | 5 | -2 | 7 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| R | -2 | 7 | -1 | -1 | 7 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| N | -1 | 1 | 0 | 0 | -3 | 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 | -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 | -3 | -3 | 0 | -2 | -3 | -1 | 5 |   |   |   |   |   |   |
| T | 0 | -1 | 0 | -1 | -1 | -1 | -1 | -2 | -2 | -1 | -1 | -1 | -2 | -1 | 2 | 5 |   |   |   |   |
| W | -3 | -3 | -4 | -5 | -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 |
**Pairwise alignments: insertions/deletions**

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-LS</td>
<td>PADKTN</td>
<td>VAAWGK</td>
<td>VGHAGEYGA</td>
<td>EALERMF</td>
</tr>
<tr>
<td>Beta</td>
<td>VHLTP</td>
<td>EESAVT</td>
<td>A</td>
<td>VLWGK</td>
<td>V-</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>QVK</td>
<td>GH</td>
<td>GKK</td>
<td>VADAL</td>
<td>TNAVAH</td>
</tr>
<tr>
<td>Beta</td>
<td>KVK</td>
<td>AHG</td>
<td>KVL</td>
<td>GAFSD</td>
<td>GLAH</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>PAEFTPAVHA</td>
<td>SLDKFLASV</td>
<td>TSTV</td>
<td>LTSKYR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beta</td>
<td>GKEFTPPVQA</td>
<td>AYQKVVA</td>
<td>G</td>
<td>VANA</td>
<td>LAHK</td>
<td>YH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>120</th>
<th>130</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beta</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Alignment scores: insertions/deletions

Affine gap penalties:
Multiple insertions/delections 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  

Note: fake matrix constructed for pedagogic purposes.
### Handout: summary of results

<table>
<thead>
<tr>
<th></th>
<th>Alignment 1</th>
<th>Alignment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOSUM-50</td>
<td>38</td>
<td>51</td>
</tr>
<tr>
<td>BLOSUM-Trp</td>
<td>118</td>
<td>91</td>
</tr>
</tbody>
</table>
# Protein substitution matrices

## 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)

- **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
Estimation of the BLOSUM 50 matrix

- BLOSUM matrices are computed based on gap-free alignments in the so-called BLOCKS database. BLOSUM 50 is computed by comparing sequences that are less than 50% identical. BLOSUM 80 is computed from sequences less than 80% identical, etc.

- All pairs of sequences in a block are compared, and the observed pair frequencies ($p_{ab}$) are noted. For instance: $p_{WW} = 0.0065$, $p_{AL} = 0.0044$, etc.

- Expected pair frequencies are computed from single amino acid frequencies. For instance: $p_A \times p_L = 0.074 \times 0.099 = 0.0073$

- For each amino acid pair the substitution scores are essentially computed as follows (here $\lambda$ is a scaling factor, used to obtain integer scores):

$$
S(a, b) = \frac{1}{\lambda} \ln \left( \frac{p_{obs}}{p_{exp}} \right) = \frac{1}{\lambda} \ln \left( \frac{p_{ab}}{p_a \times p_b} \right)
$$
Estimation of the BLOSUM 50 matrix

• Example: $A + L$:

\[
p_A = 0.074, p_L = 0.099 \Rightarrow p_A \times p_L = 0.0073
\]

\[
p_{AL} = 0.044
\]

\[
S_{AL} = \frac{1}{\lambda} \ln \left( \frac{p_{AL}}{p_A \times p_L} \right)
\]

\[
= \frac{1}{0.347} \ln \left( \frac{0.0044}{0.0073} \right)
\]

\[
= -0.51
\]

\[
= \frac{0.347}{0.347}
\]

\[
= -1.46
\]

\[
\approx -1
\]
Optimal alignment:

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

So:

best alignment can be found by exhaustively searching all possible alignments, scoring each of them and choosing the one with the highest score?
How many possible alignments are there?

Handout exercise: Enumerate all the possible alignments for two sequences of lengths $n_1=2$ and $n_2=3$
How many possible alignments are there?
Derivation of formula

- **Starting point:** Two sequences, \(s_1\) and \(s_2\)
  - Lengths: \(n_1\) and \(n_2\), where \(s_1\) is the longest \((n_1 > n_2)\)
How many possible alignments are there? Derivation of formula

- **Starting point**: Two sequences, $s_1$ and $s_2$
  - Lengths: $n_1$ and $n_2$, where $s_1$ is the longest ($n_1 > n_2$)

- Example:
  
  $s_1$: ABCDE  \hspace{1cm} n_1=5
  
  $s_2$: 123  \hspace{1cm} n_2=3
How many possible alignments are there? Derivation of formula

- **Step 1:** We want to add k gaps to $s_1$
  - Note: $k \leq n_2$ (we don’t want to align gap with gap)
  - The total length of the alignment will then be: $n_1+k$
How many possible alignments are there? Derivation of formula

- **Step 1:** We want to add $k$ gaps to $s_1$
  - Note: $k \leq n_2$ (we don’t want to align gap with gap)
  - The total length of the alignment will then be: $n_1+k$

$$k=2 \implies n_1+k = 7$$

(potential alignment positions shown as empty boxes below)

$s_1$: ABCDE

$s_2$: 123
How many possible alignments are there?
Derivation of formula

- **Step 2:** Place the \(n_2\) non-gap symbols of \(s_2\) in the \(n_1+k\) boxes
  - The number of ways of doing this is given by the following binomial coefficient:

\[
N = \binom{n_1 + k}{n_2} = \frac{(n_1 + k)!}{(n_2)!(n_1 + k - n_2)!}
\]

\(s_1:\) ABCDE
\(s_2:\) 123
How many possible alignments are there?

Derivation of formula

- **Step 2:** Place the $n_2$ non-gap symbols of $s_2$ in the boxes
  - The number of ways of doing this is given by the following binomial coefficient:

$$N = \binom{n_1 + k}{n_2} = \frac{(n_1 + k)!}{(n_2)!(n_1 + k - n_2)!}$$

$s_1$: ABCDE

$s_2$:  

\[
\begin{array}{|c|c|c|c|}
\hline
& 1 & 2 & 3 \\
\hline
\end{array}
\]

$$N = \frac{7!}{3! \times 4!} = 35$$
How many possible alignments are there?
Derivation of formula

- **Step 2:** Place the $n_2$ non-gap symbols of $s_2$ in the boxes
  - The number of ways of doing this is given by the following binomial coefficient:

  \[
  N = \binom{n_1 + k}{n_2} = \frac{(n_1 + k)!}{(n_2)!(n_1 + k - n_2)!}
  \]

  \[
  s_1: \text{ABCDE} \quad \begin{array}{cccc}
  & & & \\
  & & & \\
  \end{array}
  \]

  \[
  s_2: \quad - & - & 1 & - & 2 & 3 & -
  \]

  \[
  N = \frac{7!}{3! \times 4!} = 35
  \]

  - The remaining boxes will then contain gap symbols
How many possible alignments are there? Derivation of formula

- **Step 3:** Place the $k$ gaps in $s_1$ opposite the $n_2$ non-gap symbols in $s_2$ (this ensures that gaps in $s_1$ are never aligned with a gap in $s_2$)

- The number of ways of doing this is given by the following binomial coefficient:

$$N = \binom{n_2}{k} = \frac{(n_2)!}{k!(n_2 - k)!}$$

$s_1$: ABCDE

$s_2$: - - 1 - 2 3 -
How many possible alignments are there? Derivation of formula

- **Step 3:** Place the k gaps in $s_1$ opposite the $n_2$ non-gap symbols in $s_2$ (this ensures that a gap in $s_1$ never is aligned with a gap in $s_2$)

- The number of ways of doing this is given by the following binomial coefficient:

$$N = \binom{n_2}{k} = \frac{(n_2)!}{k!(n_2 - k)!}$$

$s_1$: ABCDE

$s_2$:  

|   | - | - | 1 | - | 2 | 3 | - |

$N = \binom{3}{2} = \frac{3!}{2!1!} = 3$
How many possible alignments are there?

Derivation of formula

- **Step 3:** Place the k gaps in s₁ opposite the n₂ non-gap symbols in s₂ (this ensures that gaps in s₁ are never aligned with a gap in s₂)

- The number of ways of doing this is given by the following binomial coefficient:

\[
N = \binom{n_2}{k} = \frac{(n_2)!}{k!(n_2 - k)!}
\]

<table>
<thead>
<tr>
<th>s₁:</th>
<th>A</th>
<th>B</th>
<th>-</th>
<th>C</th>
<th>D</th>
<th>-</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>s₂:</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>2</td>
<td>3</td>
<td>-</td>
</tr>
</tbody>
</table>

\[
N = \binom{3}{2} = \frac{3!}{2!1!} = 3
\]

- The remaining boxes will then contain the non-gap symbols, in order
How many possible alignments are there?
Derivation of formula

• For k gaps added to $s_1$ we therefore have:
  
  • Number of ways of placing the $n_2$ non-gap symbols from $s_2$ in the $n_1+k$ “alignment boxes”:

    $$N = \binom{n_1 + k}{n_2}$$

    \[
    \begin{array}{ccccccc}
    & & & & & & \\
    - & - & 1 & - & 2 & 3 & - \\
    \end{array}
    \]

  • Number of ways of placing the k gap symbols from $s_1$ opposite from the $n_2$ non-gap symbols in this alignment:

    $$N = \binom{n_2}{k}$$

    \[
    \begin{array}{ccccccc}
    A & B & - & C & D & - & E \\
    - & - & 1 & - & 2 & 3 & - \\
    \end{array}
    \]

  • The total number of alignments with k gaps added to $s_1$ is therefore:

    $$N = \binom{n_1 + k}{n_2} \binom{n_2}{k}$$
How many possible alignments are there? Derivation of formula

- For each possible value of \( k \) the number of alignments is given by:

\[
N = \binom{n_1 + k}{n_2} \binom{n_2}{k}
\]

- Given that \( k \) can be any number from 0 to \( n_2 \), the total number of possible alignments of sequences of length \( n_1 \) and \( n_2 \) is therefore:

\[
N = \sum_{k=0}^{n_2} \binom{n_1 + k}{n_2} \binom{n_2}{k}
\]
How many possible alignments are there? Derivation of formula

- Example: \( n_1=3, \ n_2=2:\)

\[
N = \sum_{k=0}^{2} \binom{3+k}{2} \binom{2}{k}
\]

\[
= \binom{3}{2} \binom{2}{0} + \binom{4}{2} \binom{2}{1} + \binom{5}{2} \binom{2}{2}
\]

\[
= 3 \times 1 + 6 \times 2 + 10 \times 1
\]

\[
= 3 + 12 + 10
\]

\[
= 25
\]
How many possible alignments are there?

<table>
<thead>
<tr>
<th>Length of sequences: $n_1 = n_2$</th>
<th>Number of possible alignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>63</td>
</tr>
<tr>
<td>4</td>
<td>321</td>
</tr>
<tr>
<td>5</td>
<td>1683</td>
</tr>
<tr>
<td>10</td>
<td>8,097,453</td>
</tr>
<tr>
<td>20</td>
<td>$2.61 \times 10^{14}$</td>
</tr>
<tr>
<td>100</td>
<td>$2.05 \times 10^{75}$</td>
</tr>
<tr>
<td>300</td>
<td>$1.53 \times 10^{228}$</td>
</tr>
</tbody>
</table>
The number of possible pairwise alignments increases explosively with the length of the sequences:

\[ N = \sum_{k=0}^{n_2} \binom{n_1 + k}{n_2} \binom{n_2}{k} \]

Two protein sequences of length 300 amino acids can be aligned in approximately $10^{228}$ different ways.

Time needed to test all possibilities much larger than the entire lifetime of the universe.
Pairwise alignment: the solution

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

\[
\begin{array}{cccccc}
T & C & G & C & A \\
T & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
C & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
C & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
A & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\end{array}
\]

\[
\Rightarrow \quad \text{TCGCA} \\
\text{TC-CA}
\]

\[
\begin{array}{cccccc}
T & C & G & C & A \\
T & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
C & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
C & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
A & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\end{array}
\]

\[
\Rightarrow \quad \text{TCGCA} \\
\text{T-CCA}
\]
Any given point in matrix can only be reached from three possible previous 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”).

\[ \text{score}(x,y) = \max \left\{ \text{score}(x,y-1) - \text{gap-penalty} \right\} \]

\[ \text{score}(x,y) = \max \left\{ \text{score}(x,y-1) - \text{gap-penalty} \right\} \]
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.

\[
\text{score}(x,y) = \max \left\{ \begin{array}{l}
\text{score}(x,y-1) - \text{gap-penalty} \\
\text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\
\text{score}(x-1,y) - \text{gap-penalty}
\end{array} \right. 
\]
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.

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

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

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>G</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Gaps: -2
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

```
0  0
-2 -4
1  -6
-4 -8
-6 -10
T C t[j] G C A
```

-3 -6
-1

s[i]

C2

C3

A4
Dynamic programming: example
Dynamic programming: example

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

\[
\begin{array}{cccccc}
 & T & C & G & C & A \\
0 & 0 & -2 & -4 & -6 & -8 & -10 \\
1 & -2 & 1 & -1 & -3 & -5 & -7 \\
2 & -4 & -1 & 2 & 0 & -2 & -4 \\
3 & -6 & -3 & 0 & 1 & 1 & -1 \\
4 & -8 & -5 & -2 & -1 & 0 & 2 \\
\end{array}
\]

\[
\begin{align*}
T & : : : :
C & : : : :
G & : : : :
C & : : : :
A & : : : :
\hline
\text{T C G C A} & 1+1-2+1+1 = 2
\end{align*}
\]
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 a 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.