**Pairwise alignment**

**Main goal:** to identify genes and regulatory elements in genomic sequences by comparing to known genes in related organisms

**Evolution:** introduces incremental changes in genetic code – these changes lead to formation of new species

- **Substitution**
  
  original: GATTAGCTCGGGCT  
  new: GATTAGCTCCGGCT  

- **Insertion**
  
  original: GATTAGCTC---GGGCT  
  new: GATTAGCTCAGGCT

- **Deletion**
  
  original: GATTAGCTCGGGCT  
  new: GATTAG--CGGGCT

**Basic problem:** Given two sequences find out whether they are related – requires scoring and alignment

**Alignment:** relative shift of sequences or insertion of gaps in order to maximize the score

**Computation of score** uses substitution matrix $p_{xy}$

Sequences $x = x_1 x_2 ... x_m$, $y = y_1 y_2 ... y_m$

**Score (log odds ratio):** $S = \sum_i s(x_i, y_i)$

where $s(a, b)$ is the substitution matrix $s(a, b) = \log \left( \frac{p_{ab}}{q_a q_b} \right)$

**Substitution matrix:** gives log-odds ratio for each aligned pair of letters

DNA: Jukes-Cantor, Felsenstein, Kimura 2-parameter

Proteins: BLOSUM62 (standard for ungapped regions), BLOSUM50 (better for gapped regions)
BLOSUM62

Contains observed substitutions found in a broad sampling of aligned segments of polypeptides, includes blocks whose members shared at least 62% identity with any other member of that block

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>P</th>
<th>A</th>
<th>G</th>
<th>N</th>
<th>D</th>
<th>E</th>
<th>Q</th>
<th>H</th>
<th>R</th>
<th>K</th>
<th>M</th>
<th>I</th>
<th>L</th>
<th>V</th>
<th>F</th>
<th>Y</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>9</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>S</td>
<td>-1</td>
<td>4</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>1</td>
<td>4</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>P</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-7</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-4</td>
<td>-3</td>
<td>-5</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
</tr>
<tr>
<td>G</td>
<td>-3</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>N</td>
<td>-3</td>
<td>1</td>
<td>0</td>
<td>-2</td>
<td>-3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>D</td>
<td>-3</td>
<td>0</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>E</td>
<td>-4</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>Q</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>H</td>
<td>-3</td>
<td>-1</td>
<td>0</td>
<td>-2</td>
<td>-2</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>2</td>
<td>-2</td>
</tr>
<tr>
<td>R</td>
<td>-3</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>-2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>-1</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>K</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
</tr>
<tr>
<td>M</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>I</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>-4</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>L</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-1</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>V</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>F</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-4</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-3</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-3</td>
<td>1</td>
</tr>
</tbody>
</table>
| W | -2 | -3 | -3 | -4 | -3 | -2 | -4 | -4 | -3 | -1 | -2 | -1 | -1 | -1 | -1 | -3 | -2 | -3 |  1 | -1 |  3 | -1 |  2 |  4 | 11
**Gap penalty:** cost for including gap in an alignment

\[ \gamma(g) = -d - (g - 1)e \]

- \( g \) length of the gap
- \( \gamma \) gap penalty score
- \( d \) gap-opening penalty
- \( e \) gap-extension penalty

Standard values: \( d = 12, \ e = 2 \) or \( d = 8, \ e = 0 \)

Probability of occurrence of a gap

\[ P(\text{gap}) = f(g) \prod_{i \in \text{gap}} q_{x_i} \]

\[ \gamma(g) = \log(f(g)) \]

**Problem:**

Using the BLOSUM62 substitution matrix and gap penalty with \( d = 12, \ e = 2 \) calculate the score for the following alignment

\( x: \) NALSALSD----LHAHKL

\( y: \) AANAALLDEFPQFKAHQE

**Ans:**

**Alignment algorithm:** finds the alignment that maximizes the score

Score is additive (including the gap penalty) therefore we can use dynamic programming algorithms.

Four types of alignment:

- global (Needleman-Wunsch algorithm)
- local (Smith-Waterman algorithm)
- repeated matches
- overlap matches
Global Alignment

**Idea:** Build optimal alignment using solutions of optimal alignments for smaller sequences

**Procedure:** Recursively construct a matrix $F$ in which $F(i, j)$ is the score of the optimal alignment of sequences $x_1 x_2 \ldots x_i$ and $y_1 y_2 \ldots y_j$.

\[ F(0,0) = 0 \]
\[
F(i, j) = \max \left( \begin{array}{c}
F(i-1, j-1) + s(x_i, y_j) \\
F(i-1, j) - d \\
F(i, j-1) - d
\end{array} \right) \quad \text{for } x_i \text{ aligned to } y_j \]
\[
F(i, j) = \max \left( \begin{array}{c}
F(i-1, j) - d \\
F(i, j-1) - d
\end{array} \right) \quad \text{for } x_i \text{ aligned to a gap}
\]

(keep track of which of the alternatives produced the maximum)

The end of the aligned sequence is the bottom right corner.

Perform traceback.

**Example:** Compute the best global alignment of HEAGAWGHEE and PAWHEAE.

BLOSUM62, $d = 8$

<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>-8</td>
<td>-16</td>
<td>-24</td>
<td>-32</td>
<td>-40</td>
<td>-48</td>
<td>-56</td>
<td>-64</td>
<td>-72</td>
</tr>
<tr>
<td>A</td>
<td>-6</td>
<td>-2</td>
<td>-9</td>
<td>-17</td>
<td>-25</td>
<td>-33</td>
<td>-41</td>
<td>-49</td>
<td>-57</td>
<td>-65</td>
</tr>
<tr>
<td>W</td>
<td>-16</td>
<td>-10</td>
<td>-3</td>
<td>-5</td>
<td>-13</td>
<td>-21</td>
<td>-29</td>
<td>-37</td>
<td>-45</td>
<td>-53</td>
</tr>
<tr>
<td>H</td>
<td>-24</td>
<td>-18</td>
<td>-11</td>
<td>-6</td>
<td>-7</td>
<td>-15</td>
<td>-22</td>
<td>-29</td>
<td>-37</td>
<td>-45</td>
</tr>
<tr>
<td>E</td>
<td>-32</td>
<td>-16</td>
<td>-10</td>
<td>-13</td>
<td>0</td>
<td>-9</td>
<td>-17</td>
<td>-12</td>
<td>-10</td>
<td>-10</td>
</tr>
<tr>
<td>E</td>
<td>-48</td>
<td>-32</td>
<td>-19</td>
<td>-7</td>
<td>-15</td>
<td>-11</td>
<td>-12</td>
<td>-12</td>
<td>-20</td>
<td>-13</td>
</tr>
</tbody>
</table>

Notice the alternative alignments for the first and second half:

HEAGA

WGHEE

-P-

-PA--

-W-HEAE

WHEAE

--P--A

-HEAGAWGHEE

-PAWHEAE
Here only the first half has alternative alignments

HEAGA
-PA--
-P--A
--P-A
Local alignment

Idea: Only subsequences of the given sequences may properly match (e.g., two protein sequences sharing a common domain)

Do not allow for negative scores – replace them with 0 to indicate the start of possible local alignment

Procedure: Recursively construct a matrix $F$ in which $F(i, j)$ is the score of the optimal alignment of sequences $x_1x_2...x_i$ and $y_1y_2...y_j$.

\[
F(0,0) = 0
\]

\[
F(i, j) = \max \begin{cases} 
0 & x_i \text{ aligned to } y_j \\
F(i-1, j-1) + s(x_i, y_j) & x_i \text{ aligned to } y_j \\
F(i-1, j-1) - d & x_i \text{ aligned to a gap} \\
F(i-1, j) - d & y_j \text{ aligned to a gap}
\end{cases}
\]

(keep track of which of the alternatives produced the maximum)

The end of the aligned sequence is the highest score over the entire matrix.

Perform traceback until you reach 0.

Example: Compute the best local alignment of HEAGAWGHEE and PAWHEAE.

BLOSUM62, $d = 8$
Repeated matches

Idea: A subsequence of the one sequence may occur several times in the other sequence (e.g., binding domain of a protein in a regulatory region)

Do not allow for negative scores and include threshold $T$

Asymmetric procedure: finds one or more non-overlapping sections of sequence $y$ (the motif) in sequence $x$

Procedure: Recursively construct a matrix $F$ in which $F(i,j)$ is the score of the optimal alignment of sequences $x_1x_2...x_i$ and $y_1y_2...y_j$.

$$F(0,0) = 0$$

$$F(i,0) = \max \begin{pmatrix} F(i-1,0) \\ F(i-1,j) - T \end{pmatrix}$$

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

(keep track of which of the alternatives produced the maximum)

The end of the aligned sequence is at the first element of the last column.
Perform traceback until you reach 0.
Algorithm obtains all local matches in one pass.

Example: Compute the repeated matches of PAWHEAE in HEAGAWGHEE.

Result:

HEAGAWGHEE
HEA.AW-HE.
Overlap Match

**Idea:** No penalty for gaps at the ends of sequences (e.g., matching of fragments of shotgun sequencing)

**Procedure:** Recursively construct a matrix $F$ in which $F(i, j)$ is the score of the optimal alignment of sequences $x_1x_2...x_i$ and $y_1y_2...y_j$.

$$F(i, 0) = 0$$
$$F(0, j) = 0$$

$$F(i, j) = \max \begin{cases} 
F(i - 1, j - 1) + s(x_i, y_j) & \text{if } x_i \text{ aligned to } y_j \\
F(i - 1, j) - d & \text{if } x_i \text{ aligned to a gap} \\
F(i, j - 1) - d & \text{if } y_j \text{ aligned to a gap}
\end{cases}$$

(keep track of which of the alternatives produced the maximum)

The end of the aligned sequence is the highest score over the last row and column.

Perform traceback until you reach 0.

**Example:** Compute the overlap match of PAWHEAE and HEAGAWGHEE.

**Result:**

GAWGHEE

PAW-HEA
Global alignment with affine gap scores

Idea: Replace $F(i, j)$ with

$M(i, j)$ - the best score up to $(i, j)$ position given that $x_i$ matches with $y_j$

$I_x(i, j)$ - the best score up to $(i, j)$ position given that $x_i$ aligns with a gap

$I_y(i, j)$ - the best score up to $(i, j)$ position given that $y_j$ aligns with a gap

Gap penalty: $\gamma(g) = -d - (g - 1)e$

Procedure: Recursively construct $M(i, j), I_x(i, j), I_y(i, j)$

$M(0, 0) = I_x(0, 0) = I_y(0, 0)$

$M(i, j) = \max \left\{ M(i - 1, j - 1) + s(x_i, y_j), I_x(i - 1, j - 1) + s(x_i, y_j), I_y(i - 1, j - 1) + s(x_i, y_j) \right\}$

$I_x(i, j) = \max \left\{ M(i - 1, j) - d, I_x(i - 1, j) - e \right\}$

$I_y(i, j) = \max \left\{ M(i, j - 1) - d, I_y(i, j - 1) - e \right\}$
Deriving BLOSUM matrices
[Henikoff & Henikoff, PNAS 89 (1992): 10915-10919.]

- A set of ungapped aligned regions is aligned, clustered at 62% (or 50%) level
- Frequencies $A_{ab}$ of observing residue $a$ in one cluster aligned with residue $b$ in another cluster are calculated
- $q_a$ and $p_{ab}$ are estimated

$$q_a = \frac{\sum_b A_{ab}}{\sum_{cd} A_{cd}} \quad p_{ab} = \frac{A_{ab}}{\sum_{cd} A_{cd}}$$

- The matrix $s(a,b)$ is calculated by $s(a,b) = \log \left( \frac{p_{ab}}{q_a q_b} \right)$