Sequence Alignment Algorithms

Morten Nielsen
BioSys, DTU
Why learn about alignment algorithms?

- All publicly available alignment programs do the same thing
  - Sequence alignment using amino acids substitution matrices and affine gap penalties
- This is fast but not optimal
  - Protein alignment is done much more accurate using sequence profiles and position specific gap penalties (price for gaps depends on the structure)
  - Must implement your own alignment algorithm to do this
Outline

- What you have been told is not true -)
  - Alignment algorithms are more complex
- The true sequence alignment algorithm story
  - The slow algorithm (O3)
  - The fast algorithm (O2)
Sequence alignment
The old Story
Pairwise alignment: the solution

"Dynamic programming"
(the Needleman-Wunsch algorithm)

Match score = 2
Gap penalty = -2
Alignment depicted as path in matrix

Match score = 2
Gap penalty = -2

TCGCA
TC-CA
Score=2

TCGCA
T-CCA
Score=0
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

---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.

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

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

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

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.

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

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

Gaps: -2
Dynamic programming: example

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

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

Gaps: -2
Dynamic programming: example

```
A  C  G  T
A  1 -1 -1 -1
C -1  1 -1 -1
G -1 -1  1 -1
T -1 -1 -1  1
```

Gaps: -2
Dynamic programming: example

\[
\begin{array}{cccccc}
& T & C & G & C & A \\
0 & 0 & -2 & -4 & -6 & -8 & -10 \\
T1 & -2 & 1 & -1 & -3 & -5 & -7 \\
C2 & -4 & -1 & 2 & 0 & -2 & -4 \\
C3 & -6 & -3 & 0 & 1 & 1 & -1 \\
A4 & -8 & -5 & -2 & -1 & 0 & 2 \\
\end{array}
\]

\[
T C G C A \\
\underline{1+1-2+1+1} = 2
\]
A now the truth
Dynamic programming: computation of scores

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

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: example

What about $j-2$ and a gap extension?

$$a[i,j] = \max \begin{cases} 
  a[i,j-1] - 2 \\
  a[i-1,j-1] + p(i,j) \\
  a[i-1,j] - 2 
\end{cases}$$
And now the true algorithm

\[
D_{m,n} = \text{Max} \begin{cases} 
D_{m+1,n+1} + d(m,n), \text{match} \\
\text{Max}_{1 \leq k \leq N-m} \left[ D_{m+k,n} + w_k \right] \\
\text{Max}_{1 \leq k \leq N-n} \left[ D_{m,n+k} + w_k \right] \\
0 \text{ (for local alignment)}
\end{cases}
\]

Start from here
How does it work (score matrix d)?

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>-3</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>P</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
<td>10</td>
</tr>
</tbody>
</table>

**d matrix**

**Blosum 50 matrix**
How does it work? (The slow way O3)

D matrix

$$D_{m,n} = \begin{cases} D_{m+1,n+1} + \alpha(m,n), & \text{match} \\ \text{Max}_{1 \leq k \leq N_{-m}} [D_{m+k,n} + \beta_k] \\ \text{Max}_{1 \leq k \leq N_{-n}} [D_{m,n+k} + \beta_k] \\ 0 \text{ (for local alignment)} \end{cases}$$

Gap-open $$W_1 = -5$$
Gap-extension $$U = -1$$

Start from here!
How does it work? (The slow way O3)

- Check all positions in (green) row and column to check score for gap extension.
- CPU intensive (O³)

### D matrix

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td>17</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td>10</td>
<td>15</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### d matrix

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>-3</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>P</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
<td>10</td>
</tr>
</tbody>
</table>

Gap-open $W_1 = -5$
Gap-extension $U = -1$

$$D_{m,n} = \begin{cases} 
D_{m+1,n+1} + d(m,n), \text{match} \\
\max_{1 \leq k \leq N-m} \left[ D_{m+k,n} + w_k \right] \\
\max_{1 \leq k \leq N-n} \left[ D_{m,n+k} + w_k \right] \\
0 \text{ (for local alignment)}
\end{cases}$$
How does it work? Fill out the D matrix

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>20</td>
<td></td>
<td>14</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>11</td>
<td>17</td>
<td>10</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td>9</td>
<td>10</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td></td>
<td>4</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Check all positions in (green) row and column to check score for gap extension.
- CPU intensive (O3)

**d matrix**

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>-3</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>-4</td>
</tr>
<tr>
<td>P</td>
<td>-3</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
<td>10</td>
</tr>
</tbody>
</table>

**Gap-open** \(W_1 = -5\)

**Gap-extension** \(U = -1\)

\[
D_{m,n} = \begin{cases} 
D_{m+1,n+1} + d(m,n), & \text{match} \\
\max_{1\leq k\leq N-m} \left[D_{m+k,n} + w_k\right] \\
\max_{1\leq k\leq N-n} \left[D_{m,n+k} + w_k\right] \\
0 & \text{(for local alignment)}
\end{cases}
\]
And now the fast algorithm (O2)

\[ D_{m,n} = \text{Max} \begin{cases} D_{m+1,n+1} + d(m,n), & \text{match} \\ P_{m,n}, & \text{insertion in database} \\ Q_{m,n}, & \text{insertion in query} \\ 0 & \end{cases} \]

**Affine gap penalties**

\[ w_k = w_1 + u \cdot (k - 1) \]

- Open a gap
- Extending a gap
And now the fast algorithm (O2)

\[
D_{m,n} = \max\begin{cases}
    D_{m+1,n+1} + d(m,n), & \text{match} \\
    P_{m,n}, & \text{insertion in database} \\
    Q_{m,n}, & \text{insertion in query} \\
    0
\end{cases}
\]

\[
P_{m,n} = \max_{1 \leq k \leq N-m} \left[ D_{m+k,n} + w_k \right]
\]

\[
= \max \left[ D_{m+1,n} + w_1, \max_{1 \leq k \leq N-m-1} \left( D_{m+1+k,n} + w_{k+1} \right) \right]
\]

\[
= \max \left[ D_{m+1,n} + w_1, \max_{1 \leq k \leq N-m-1} \left( D_{m+1+k,n} + w_k \right) + u \right], k + 1 > 1
\]

\[
= \max \left[ D_{m+1,n} + w_1, P_{m+1,n} + u \right]
\]

Open a gap

Extending a gap

Database (m)

Query (n)
And now the true algorithm (cont.)

\[
D_{m,n} = \begin{cases} 
D_{m+1,n+1} + d(m,n), & \text{match} \\
P_{m,n}, & \text{insertion in database} \\
Q_{m,n}, & \text{insertion in query} \\
0 & 
\end{cases}
\]

\[
P_{m,n} = \text{Max} \left[ D_{m+1,n} + w_1, P_{m+1,n} + u \right]
\]

\[
Q_{m,n} = \text{Max} \left[ D_{m,n+1} + w_1, Q_{m,n+1} + u \right]
\]
How does it work (D, Q, and P-matrices)

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

\[
P_{m,n} = \text{Max}[D_{m+1,n} + w_1, P_{m+1,n} + u]\]

\[
P_{m,n} = \text{Max}[5 - 5, -5 - 1] = 0
\]

\[
W_1 = -5
\]

\[
U = -1
\]
How does it work (D, Q, P, and E-matrices)

\[
P_{m,n} = \text{Max}[D_{m-1,n} + w_1, P_{m-1,n} + u]
\]

\[
P_{m,n} = \text{Max}[5 - 5, -5 - 1] = 0
\]

\[
W_1 = -5
\]

\[
U = -1
\]
How does it work (D, Q, and P-matrices)

\[ Q_{m,n} = \text{Max}\left[ D_{m,n+1} + w_1, Q_{m,n+1} + u \right] \]

\[ Q_{m,n} = \text{Max}\left[ 5 - 5, -5 - 1 \right] = 0 \]
How does it work (D, Q, P, and E-matrices)

\[
Q_{m,n} = \text{Max}[D_{m,n+1} + w_1, Q_{m,n+1} + u]
\]

\[
Q_{m,n} = \text{Max}[5 - 5, -5 - 1] = 0
\]
How does it work (D, Q, and P-matrices)

\[
D_{m,n} = \max \begin{cases} 
D_{m+1,n+1} + d(m,n), & \text{match} \\
P_{m,n}, & \text{insertion in database} \\
Q_{m,n}, & \text{insertion in query} 
\end{cases}
\]

\[
D_{m,n} = \max \begin{cases} 
10 + 5 \\
0 \\
0
\end{cases} = 15
\]
How does it work (D, Q, and P-matrices)

\[ D_{m,n} = \max \begin{cases} D_{m+1,n+1} + d(m,n), & \text{match} \\ P_{m,n}, & \text{insertion in database} \\ Q_{m,n}, & \text{insertion in query} \end{cases} \]

\[ D_{m,n} = \max \begin{cases} 10 + 5 \\ 0 \end{cases} = 15 \]
How does it work. Eij-matrix.
(Keeping track on the path)

$e_{ij} = 1$ match
$e_{ij} = 2$ gap-opening database
$e_{ij} = 3$ gap-extension database
$e_{ij} = 4$ gap-opening query
$e_{ij} = 5$ gap-extension query
How does it work (D,Q,P, and E-matrices)

<table>
<thead>
<tr>
<th>D</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>5</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>-5</td>
<td>-5</td>
<td>-5</td>
<td>-5</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
How does it work (D,Q,P, and E-matrices)

<table>
<thead>
<tr>
<th>D</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th>m</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>20</td>
<td>18</td>
<td>14</td>
<td>9</td>
<td>3</td>
<td>0</td>
<td>V</td>
<td>11</td>
<td>10</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>L</td>
<td>11</td>
<td>15</td>
<td>17</td>
<td>10</td>
<td>4</td>
<td>0</td>
<td>L</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>n</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>15</td>
<td>5</td>
<td>0</td>
<td>L</td>
<td>2</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>15</td>
<td>5</td>
<td>0</td>
<td>P</td>
<td>0</td>
<td>-5</td>
<td>-5</td>
<td>-5</td>
</tr>
<tr>
<td>P</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>P</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Q</th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>11</td>
<td>10</td>
<td>12</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>8</td>
<td>4</td>
<td>5</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>L</td>
<td>2</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>-5</td>
<td>-5</td>
<td>-5</td>
<td>-5</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
And the alignment

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>L</th>
<th>P</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**VLILP**

**VL-LP**
And the alignment. Gap extensions

D-matrix

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>A</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>19</td>
<td>13</td>
<td>17</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>9</td>
<td>14</td>
<td>12</td>
<td>13</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>n</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

E-matrix

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>A</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>n</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>P</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

0   0   0   0   0   0   0
And the alignment

15 - 5 - 1 = 9?
And the alignment

\[
\begin{array}{ccccccc}
D & V & L & I & A & L & P \\
V & 19 & 13 & 17 & 10 & 9 & 3 & 0 \\
L & 9 & 14 & 12 & 13 & 10 & 4 & 0 \\
n & 7 & 8 & 9 & 10 & 15 & 5 & 0 \\
P & 1 & 2 & 3 & 4 & 5 & 10 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\quad \begin{array}{ccccccc}
V & 1 & 1 & 1 & 1 & 3 & 3 & 0 \\
L & 1 & 1 & 1 & 1 & 1 & 3 & 0 \\
L & 5 & 1 & 5 & 4 & 1 & 2 & 0 \\
P & 5 & 5 & 5 & 5 & 4 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

5 - 5 - 1 = 9?
And the alignment

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>A</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>19</td>
<td>13</td>
<td>17</td>
<td>10</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>9</td>
<td>14</td>
<td>12</td>
<td>13</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>L</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>P</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>L</th>
<th>I</th>
<th>A</th>
<th>L</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>P</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

VLIALP
VL--LP

15 -5 -1 = 9? ***
5 -5 -1 = 9?
And now you!
Summary

- Alignment is more complicated than what you have been told.
- Simple algorithmic tricks allow for alignment in \( O^2 \) time.
- More heuristics to improve speed:
  - Limit gap length
  - Look for high scoring regions