Sequence alignment (continued)
Various types
Global Alignment (review)

- **Input:** Two strings, labeled $s$ and $t$
  - $|s|$ is $n$
  - $|t|$ is $m$

- **Output:** Two strings $s_A$ and $t_A$ such that:
  - $s_A$ and $t_A$ are of equal length $L$
  - Characters must be in same order, with “-” spacers as needed
  - If $s_A[i] = '-'$, then $t_A[i] \neq '-'$
  - If $t_A[i] = '-'$, then $s_A[i] \neq '-'$
Visual alignments

**Score:** A measure of alignment quality

```
C A T  -  T C A  -  C
C  -  T C G C A G C
--------------------------------
10  -5 10  -5  -2 10 10  -5 10
```

Total = 33

Scored as $E(C,C)$, $E(A,-)$, $E(T,T)$, $E(-,C)$, etc.
Types of scores

• Reasons:
  – Sequencing error(s)
  – Evolutionary change

• Three parameters
  – Gap (indel?)
  – Mismatch (misread base?)
  – Match (no change)

• We will only consider constant gap penalties for now.
Optimal alignment

• We want to find an alignment that optimizes an evaluation function $E(s_A, t_A)$

• The “brute force” method is a hard problem (NP-hard) requiring exponential time.

• Dynamic programming, in cases we will discuss, can be used to solve this problem efficiently.
Requirements

• We will need four things to compute a global alignment:
  1. Substitution matrix (parameters)
  2. Recurrence relation
  3. Filling up a table
  4. Traceback
We can solve this based on looking at three smaller problems

\[
T[i,j] = \max \left\{ T[i-1,j-1] + \text{score}(s[i],t[j]), T[i-1,j] + g, T[i,j-1] + g \right\}
\]
<table>
<thead>
<tr>
<th></th>
<th>λ</th>
<th>C</th>
<th>T</th>
<th>C</th>
<th>G</th>
<th>C</th>
<th>A</th>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>λ</td>
<td>0</td>
<td>-5</td>
<td>-10</td>
<td>-15</td>
<td>-20</td>
<td>-25</td>
<td>-30</td>
<td>-35</td>
<td>-40</td>
</tr>
<tr>
<td>C</td>
<td>-5</td>
<td>10</td>
<td>5</td>
<td>-15</td>
<td>-20</td>
<td>-25</td>
<td>-30</td>
<td>-35</td>
<td>-40</td>
</tr>
<tr>
<td>A</td>
<td>-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>-25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-35</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

+10 for match, -2 for mismatch, -5 for space (rowwise)
Traceback yields both optimal alignments in this example
End-gap free alignment

• We often don’t want to penalize gaps at the start or end of the alignment, especially when comparing short and long sequences

• Same as global alignment, except:
  – Initialize with zeros (free gaps at start)
  – Locate max in the last row/column (free gaps at end)

• Also called *semiglobal alignment*
Example paths
Example traceback paths

-AGCC-TAG-CCAA
AGCTAAG-CT-A-A
AGCCTA-GCCAA
AGCT-AAGCTAA
AGCCTA-GCCAA
AGC-TAAGCTAA
-------------------AGCCTAGCCAA
AGCTAAGCTAA--------------
Practice

\[ T[i, 0] = 0 \quad T[0, j] = 0 \]

\[ T[i, j] = \max \begin{cases} 
T[i-1, j-1] + \text{score}(s[i], t[j]) \\
T[i-1, j] + g \\
T[i, j-1] + g 
\end{cases} \]

S = CATTA
T = ACATTAG

Find the score of the best semiglobal alignment

\[
\begin{align*}
\text{Match} &= 5 \\
\text{Mismatch} &= -2 \\
\text{Gap} &= -3
\end{align*}
\]
+10 for match, -2 for mismatch, -5 for space (rowwise)
Local Alignment

\[ T[i, j] = \text{Score of optimally aligning a suffix of } s \text{ with a suffix of } t. \]

\[
T[i, j] = \max \begin{cases} 
T[i - 1, j - 1] + \text{score}(s[i], t[j]) \\
T[i - 1, j] + g \\
T[i, j - 1] + g \\
0 
\end{cases}
\]

Initialize top row and leftmost column to zero.
Practice

\[ T[i, 0] = 0 \quad T[0, j] = 0 \]

\[
T[i, j] = \max \begin{cases} 
T[i-1, j-1] + \text{score}(s[i], t[j]) \\
T[i-1, j] + g \\
T[i, j-1] + g \\
0
\end{cases}
\]

S = CTACT
T = ATACG

Find the score of the best local alignment

Match = 5
Mismatch = -2
Gap = -3
+1 for a match, -1 for a mismatch, -5 for a gap
Reducing space requirements

- $O(mn)$ tables are often the limiting factor in computing large alignments

- There is a linear space technique that only doubles the time required
  [Hirschberg77]
Alignment and new architectures