Sequence Alignment (chapter 6)

- The biological problem
- Global alignment
- Local alignment
- Multiple alignment
Background: comparative genomics

- Basic question in biology: *what properties are shared among organisms?*
- Genome sequencing allows comparison of organisms at DNA and protein levels
- Comparisons can be used to
  - Find evolutionary relationships between organisms
  - Identify functionally conserved sequences
  - Identify corresponding genes in human and model organisms: develop models for human diseases
Homologs

• Two genes or characters $g_B$ and $g_C$ evolved from the same ancestor $g_A$ are called homologs

\[ g_A = \text{agtgtccgttaagtgccgttc} \]
\[ g_B = \text{agtgccgtaaagttgtacggtc} \]
\[ g_C = \text{ctgactggtttgtggttc} \]

• Homologs usually exhibit conserved functions

• Close evolutionary relationship $\Rightarrow$ expect a high number of homologs
Sequence similarity

- Intuitively, similarity of two sequences refers to the degree of match between corresponding positions in sequence

\[
\text{agtgccgtaaagttgtacgtc} \\
\text{ctgactgtttgtggttc}
\]

- What about sequences that differ in length?
Similarity vs homology

Sequence similarity is not sequence homology

- If the two sequences $g_B$ and $g_C$ have accumulated enough mutations, the similarity between them is likely to be low.

<table>
<thead>
<tr>
<th>#mutations</th>
<th>Sequence</th>
<th>#mutations</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><code>agtgtccgtaagtgcgttc</code></td>
<td>64</td>
<td><code>acagtccgttccggctattg</code></td>
</tr>
<tr>
<td>1</td>
<td><code>agtgtccgttataagtgcgttc</code></td>
<td>128</td>
<td><code>cagagcactacccgc</code></td>
</tr>
<tr>
<td>2</td>
<td><code>agtgtccgctttatagtgccggttc</code></td>
<td>256</td>
<td><code>cacgagtaagatagct</code></td>
</tr>
<tr>
<td>4</td>
<td><code>agtgtccgcttaagggcgttc</code></td>
<td>512</td>
<td><code>taatcggtgata</code></td>
</tr>
<tr>
<td>8</td>
<td><code>agtgtccgcttcaaggggctgt</code></td>
<td>1024</td>
<td><code>acccttatcttcttgagtt</code></td>
</tr>
<tr>
<td>16</td>
<td><code>gggccgtttcatgggggt</code></td>
<td>2048</td>
<td><code>agcgacctgccccaa</code></td>
</tr>
<tr>
<td>32</td>
<td><code>gcaggagccgctcactgaggggt</code></td>
<td>4096</td>
<td><code>caaac</code></td>
</tr>
</tbody>
</table>

Homology is more difficult to detect over greater evolutionary distances.
Similarity vs homology (2)

- Sequence similarity can occur by chance
  - *Similarity does not imply homology*

- Similarity is an expected consequence of homology
Orthologs and paralogs

We distinguish between two types of homology

- Orthologs: homologs from two different species
- Paralogs: homologs within a species

Organism B

Gene A is copied within organism A

Organism C
Orthologs and paralogs (2)

- Orthologs typically retain the original function
- In paralogs, one copy is free to mutate and acquire new function (no selective pressure)

Organism B

Organism C

Gene A is copied within organism A
### Sequence alignment

**Alignment specifies which positions in two sequences match**

<table>
<thead>
<tr>
<th>acgtctag</th>
<th>acgtctag</th>
<th>acgtctag</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>actctag-</td>
<td>-actctag</td>
<td>ac-tctag</td>
</tr>
</tbody>
</table>

- 2 matches
- 5 mismatches
- 1 not aligned

- 5 matches
- 2 mismatches
- 1 not aligned

- 7 matches
- 0 mismatches
- 1 not aligned
Mutations: Insertions, deletions and substitutions

**Indel**: insertion or deletion of a base with respect to the ancestor sequence

```
acgtctag
```

**Mismatch**: substitution (point mutation) of a single base

```
actctag
```

- Insertions and/or deletions are called *indels*
  - *We can’t tell whether the ancestor sequence had a base or not at indel position*
Problems

- What sorts of alignments should be considered?
- How to score alignments?
- How to find optimal or good scoring alignments?
- How to evaluate the statistical significance of scores?

In this course, we discuss the first three problems.

Course *Biological sequence analysis* tackles all four in-depth.
Sequence Alignment (chapter 6)

- The biological problem
- *Global alignment*
- Local alignment
- Multiple alignment
Global alignment

- Problem: find optimal scoring alignment between two sequences (Needleman & Wunsch 1970)

- We give score for each position in alignment

  - Identity (match) \( +1 \) WHAT
  - Substitution (mismatch) \( -\mu \) ||
  - Indel \( -\delta \) WH–Y

\[
S(\text{WHAT}/\text{WH-Y}) = 1 + 1 - \delta - \mu
\]
Representing alignments and scores

<table>
<thead>
<tr>
<th></th>
<th>-</th>
<th>W</th>
<th>H</th>
<th>A</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Representing alignments and scores

<table>
<thead>
<tr>
<th>WHAT</th>
<th></th>
<th>W</th>
<th>H</th>
<th>A</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-</td>
<td>W</td>
<td>H</td>
<td>A</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td>2</td>
<td>2-δ</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2-δ-μ</td>
</tr>
</tbody>
</table>

Global alignment score $S_{3,4} = 2-\delta-\mu$
Dynamic programming

- How to find the optimal alignment?
- We use previous solutions for optimal alignments of smaller subsequences
- This general approach is known as dynamic programming
Filling the alignment matrix

Consider the alignment process at shaded square.

**Case 1.** Align H against H (match or substitution).

**Case 2.** Align H in WHY against – (indel) in WHAT.

**Case 3.** Align H in WHAT against – (indel) in WHY.
Filling the alignment matrix (2)

<table>
<thead>
<tr>
<th></th>
<th>-</th>
<th>W</th>
<th>H</th>
<th>A</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>W</td>
<td></td>
<td></td>
<td>Case 1</td>
<td></td>
<td>Case 2</td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td>Case 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Scoring the alternatives.

Case 1. $S_{2,2} = S_{1,1} + s(2, 2)$

Case 2. $S_{2,2} = S_{1,2} - \delta$

Case 3. $S_{2,2} = S_{2,1} - \delta$

$s(i, j) = 1$ for matching positions,

$s(i, j) = -\mu$ for substitutions.

Choose the case (path) that yields the maximum score.

Keep track of path choices.
Global alignment: formal development

\[ A = a_1 a_2 a_3 \ldots a_n, \]
\[ B = b_1 b_2 b_3 \ldots b_m \]

\[
\begin{array}{cccc}
  b_1 & b_2 & b_3 & b_4 \\
  - & a_1 & a_2 & a_3 \\
\end{array}
\]

Any alignment can be written as a unique path through the matrix.

Score for aligning \( A \) and \( B \) up to positions \( i \) and \( j \):

\[ S_{i,j} = S(a_1 a_2 a_3 \ldots a_i, b_1 b_2 b_3 \ldots b_j) \]
Scoring partial alignments

- Alignment of $A = a_1a_2a_3\ldots a_n$ with $B = b_1b_2b_3\ldots b_m$ can end in three ways
  - Case 1: $(a_1a_2\ldots a_{i-1}) a_i$
    $(b_1b_2\ldots b_{j-1}) b_j$
  - Case 2: $(a_1a_2\ldots a_{i-1}) a_i$
    $(b_1b_2\ldots b_j)$ -
  - Case 3: $(a_1a_2\ldots a_i)$ -
    $(b_1b_2\ldots b_{j-1}) b_j$
Scoring alignments

Scores for each case:

- Case 1: \((a_1 a_2 \ldots a_{i-1}) a_i (b_1 b_2 \ldots b_{j-1}) b_j\)
  
  \[s(a_i, b_j) = \begin{cases} 
  +1 & \text{if } a_i = b_j \\
  -\mu & \text{otherwise}
  \end{cases}\]

- Case 2: \((a_1 a_2 \ldots a_{i-1}) a_i (b_1 b_2 \ldots b_j) -\)

- Case 3: \((a_1 a_2 \ldots a_i) - (b_1 b_2 \ldots b_{j-1}) b_j\)

  \[s(a_i, -) = s(-, b_j) = -\delta\]
Scoring alignments (2)

- First row and first column correspond to initial alignment against indels:
  \[ S(i, 0) = -i \delta \]
  \[ S(0, j) = -j \delta \]

- Optimal global alignment score \( S(A, B) = S_{n,m} \)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>b₁</td>
<td>b₂</td>
<td>b₃</td>
<td>b₄</td>
</tr>
<tr>
<td>1</td>
<td>a₁</td>
<td>-δ</td>
<td></td>
<td>-2δ</td>
<td>-3δ</td>
</tr>
<tr>
<td>2</td>
<td>a₂</td>
<td></td>
<td>-2δ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>a₃</td>
<td></td>
<td></td>
<td>-3δ</td>
<td></td>
</tr>
</tbody>
</table>
Algorithm for global alignment

Input sequences $A, B$, $n = |A|$, $m = |B|$

Set $S_{i,0} := -\delta i$ for all $i$

Set $S_{0,j} := -\delta j$ for all $j$

for $i := 1$ to $n$

for $j := 1$ to $m$

\[ S_{i,j} := \max\{S_{i-1,j} - \delta, S_{i-1,j-1} + s(a_i,b_j), S_{i,j-1} - \delta\} \]

end

end

Algorithm takes $O(nm)$ time and space.
Global alignment: example

\[ \mu = 1 \]
\[ \delta = 2 \]

<table>
<thead>
<tr>
<th></th>
<th>-</th>
<th>T</th>
<th>G</th>
<th>G</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>0</td>
<td>-2</td>
<td>-4</td>
<td>-6</td>
<td>-8</td>
<td>-10</td>
</tr>
<tr>
<td>A</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>-8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Global alignment: example (2)

\[
\begin{array}{cccccccc}
 & - & T & G & G & T & G \\
- & 0 & -2 & -4 & -6 & -8 & -10 \\
A & -2 & -1 & -3 & -5 & -7 & -9 \\
T & -4 & -1 & -2 & -4 & -4 & -6 \\
C & -6 & -3 & -2 & -3 & -5 & -5 \\
G & -8 & -5 & -2 & -1 & -3 & -4 \\
T & -10 & -7 & -4 & -3 & 0 & -2 \\
\end{array}
\]

\[\mu = 1\]
\[\delta = 2\]

ATCGT-

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

-TGGTG
Sequence Alignment (chapter 6)

- The biological problem
- Global alignment
  - Local alignment
- Multiple alignment
Local alignment: rationale

• Otherwise dissimilar proteins may have local regions of similarity
  -> Proteins may share a function

Human bone morphogenic protein receptor type II precursor (left) has a 300 aa region that resembles 291 aa region in TGF-β receptor (right).

The shared function here is protein kinase.
Local alignment: rationale

- Global alignment would be inadequate
- Problem: find the highest scoring local alignment between two sequences
- Previous algorithm with minor modifications solves this problem (Smith & Waterman 1981)
From global to local alignment

- Modifications to the global alignment algorithm
  - Look for the highest-scoring path in the alignment matrix (not necessarily through the matrix)
  - Allow preceding and trailing indels without penalty
Scoring local alignments

A = a_1 a_2 a_3 \ldots a_n, \ B = b_1 b_2 b_3 \ldots b_m

Let I and J be intervals (substrings) of A and B, respectively: \ I \subset A, \ J \subset B

Best local alignment score:

\[ M(A, B) = \max \{ S(I, J) : I \subset A, J \subset B \} \]

where S(I, J) is the score for substrings I and J.
Allowing preceding and trailing indels

- First row and column initialised to zero:
  \[ M_{i,0} = M_{0,j} = 0 \]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>a₁</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>a₂</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>a₃</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ b₁ b₂ b₃ \]
\[ - - a₁ \]
Recursion for local alignment

- $M_{i,j} = \max \{ \$
- $M_{i-1,j-1} + s(a_i, b_i),$
- $M_{i-1,j} - \delta,$
- $M_{i,j-1} - \delta,$
- $0 \} \$

<table>
<thead>
<tr>
<th></th>
<th>-</th>
<th>T</th>
<th>G</th>
<th>G</th>
<th>T</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</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>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>
Finding best local alignment

- Optimal score is the highest value in the matrix

$$M(A, B) = \max\{S(I, J) : I \subset A, J \subset B\}$$

$$= \max_{i,j} M_{i,j}$$

- Best local alignment can be found by backtracking from the highest value in M
Local alignment: example

```
          0 1 2 3 4 5 6 7 8 9 10
          - G G C T C A A T C A
    --- +--- +--- +--- +--- +--- +--- +--- +--- +--- +--- +---
0     -  0  0  0  0  0  0  0  0  0  0
1     A  0
2     C  0
3     C  0
4     T  0
5     A  0
6     A  0
7     G  0
8     G  0
```

ACTAACTCGG-
### Local alignment: example

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</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>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

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

### Scoring

- **Match**: +2
- **Mismatch**: -1
- **Indel**: -2

---

Introduction to bioinformatics, Autumn 2006
Non-uniform mismatch penalties

- We used uniform penalty for mismatches:
  \[ s('A', 'C') = s('A', 'G') = \ldots = s('G', 'T') = \mu \]

- Transition mutations (A->G, G->A, C->T, T->C) are approximately twice as frequent than transversions (A->T, T->A, A->C, G->T)
  - use non-uniform mismatch penalties

<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>-0.5</td>
<td>-1</td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-0.5</td>
</tr>
<tr>
<td>G</td>
<td>-0.5</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>T</td>
<td>-1</td>
<td>-0.5</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>
Gaps in alignment

- Gap is a succession of indels in alignment

  \[
  \begin{array}{ccccccc}
  C & T & - & - & - & A & A \\
  C & T & C & G & C & A & A \\
  \end{array}
  \]

- Previous model scored a length k gap as \( w(k) = -k\delta \)

- Replication processes may produce longer stretches of insertions or deletions
  - In coding regions, insertions or deletions of codons may preserve functionality
Gap open and extension penalties (2)

- We can design a score that allows the penalty opening gap to be larger than extending the gap:

  \[ w(k) = -\alpha - \beta(k - 1) \]

- Gap open cost \(\alpha\), Gap extension cost \(\beta\)

- Our previous algorithm can be extended to use \(w(k)\) (not discussed on this course)
Sequence Alignment (chapter 6)

- The biological problem
- Global alignment
- Local alignment
  - *Multiple alignment*
Multiple alignment

- Consider a set of $n$ sequences on the right
  - Orthologous sequences from different organisms
  - Paralogs from multiple duplications

- How can we study relationships between these sequences?

```plaintext
aggcgagctgcagtgctta
cgtagattgacgctgac
ttccggctgcgac
gacacggcgacgagga
agttgtgccccgacgagcgaggac
gcgggtcggtgagcgctta
aagcggccctgtgtgcccta
atgctgctgccagtgta
agtccgagcccagtgctc
agtccgagtgcc
actcgggtgc
```
Optimal alignment of three sequences

- Alignment of $A = a_1a_2...a_i$ and $B = b_1b_2...b_j$ can end either in $(-, b_j)$, $(a_i, b_j)$ or $(a_i, -)$

- $2^2 - 1 = 3$ alternatives

- Alignment of $A$, $B$ and $C = c_1c_2...c_k$ can end in $2^3 - 1$ ways: $(a_i, -, -)$, $(-, b_j, -)$, $(-, -, c_k)$, $(-, b_j, c_k)$, $(a_i, -, c_k)$, $(a_i, b_j, -)$ or $(a_i, b_j, c_k)$

- Solve the recursion using three-dimensional dynamic programming matrix: $O(n^3)$ time and space

- Generalizes to $n$ sequences but impractical with moderate number of sequences
Multiple alignment in practice

- In practice, real-world multiple alignment problems are usually solved with heuristics

- Progressive multiple alignment
  - Choose two sequences and align them
  - Choose third sequence w.r.t. two previous sequences and align the third against them
  - Repeat until all sequences have been aligned
  - Different options how to choose sequences and score alignments
Multiple alignment in practice

- Profile-based progressive multiple alignment: CLUSTALW
  - Construct a distance matrix of all pairs of sequences using dynamic programming
  - Progressively align pairs in order of decreasing similarity
  - CLUSTALW uses various heuristics to contribute to accuracy
Additional material

- R. Durbin, S. Eddy, A. Krogh, G. Mitchison: Biological sequence analysis
- Course Biological sequence analysis in Spring 2007