Lecture 4
Sequence alignment: how to discover similarities between biological sequences

Chapter 6 in Jones and Pevzner

Spring 2018
January 30, 2018
Evolution as a tool for biological insight

- “Nothing in biology makes sense except in the light of evolution” - Theodosius Dobzhansky.

- The functionality of many genes is virtually the same among many organisms: Can understand biology in simpler organisms than ourselves (“model organisms”).
Homology

- Genes in organisms A and B that have evolved from the same ancestral gene are said to be *homologs*.

- Homology between genes typically indicates conserved function.

- Sequence similarity is used to infer homology.
Sequence Comparison: Early Success Story

- In 1983 Russell Doolittle and colleagues found similarities between a cancer-causing gene from the Simian Sarcoma virus and a normal growth factor gene (PDGF).

- Finding sequence similarities with genes of known function is a common approach to infer a newly sequenced gene’s function.
The drosophila “eyeless” gene

- W. Gehring discovered that turning on the “eyeless” gene in drosophila leads to the growth of ectopic eyes.
- “eyeless” is a master control gene for eye formation (transcription factor).
A similar gene in humans

- The aniridia gene in humans has a sequence that is similar to the drosophila eyeless gene.

- Eye morphogenesis is under similar genetic control in vertebrates and insects.
### PAX6_HUMAN aligned against PAX6_DRO

<table>
<thead>
<tr>
<th></th>
<th>HSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>HSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVS</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>HSGVNQLGGVFVNGRPLPDSTRQKIVELAHSGARPCDISRILQVSNGCVS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>KILGRYYETGSIRPRAIGGSKPRVATPEVVSFIGIAKYKRECPSIFAWIRD</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>104</td>
<td>KILGRYYETGSIRPRAIGGSKPRVATPEVVSFIGIAKYKRECPSIFAWIRD</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RLLSEGVCTNDNIPSVSSINRVLRNLASEKQQMGA----------</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>139</td>
<td>RLLSEGVCTNDNIPSVSSINRVLRNLASEKQQMGA----------</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RLLQENVCTNDNIPSVSSINRVLRNLAAQKEQQSTGSGSSSTSAGNSISA</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>206</td>
<td>RLLQENVCTNDNIPSVSSINRVLRNLAAQKEQQSTGSGSSSTSAGNSISA</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>---------------SWGTR--PGWYGTSVPGQPTQ-------------</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>174</td>
<td>---------------SWGTR--PGWYGTSVPGQPTQ-------------</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NHQALQQHQQQSWPPRHYSGSWYPTSLSEIPISSAPNIASTAYASGPS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>355</td>
<td>NHQALQQHQQQSWPPRHYSGSWYPTSLSEIPISSAPNIASTAYASGPS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>-----------------------------DGCQQQE--GGGE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>185</td>
<td>-----------------------------DGCQQQE--GGGE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>LAHSLSPPNDIESLASIGHQRNCPVATEDIHLKKELDGHQSDETGSGE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>405</td>
<td>LAHSLSPPNDIESLASIGHQRNCPVATEDIHLKKELDGHQSDETGSGE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NTNISISSNGEDSDEAQMRQLKRKLRQRNRTSFTQIEQIEALEKEFERTHYP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>235</td>
<td>NTNISISSNGEDSDEAQMRQLKRKLRQRNRTSFTQIEQIEALEKEFERTHYP</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>NSNGGASNIKTEDDQARLILKRKLRQRNRTSFTNDQIDSLEKEFERTHYP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>455</td>
<td>NSNGGASNIKTEDDQARLILKRKLRQRNRTSFTNDQIDSLEKEFERTHYP</td>
<td></td>
</tr>
</tbody>
</table>
Sequence alignment

**Definition**

Given two strings \( v = v_1v_2...v_m \), \( w = w_1w_2...w_n \),

an alignment is an assignment of gaps to positions 0,...,m in \( v \), and 0,...,n in \( w \), so as to line up each letter in one sequence with either a letter, or a gap in the other sequence.
Mutations at the DNA level

SEQUENCE EDITS

- Deletion
- Substitution

REARRANGEMENTS

- Inversion
- Translocation
- Duplication
Scoring an alignment

• A simple scoring scheme:
  • Penalize mismatches by $-\mu$
  • Penalize indels by $-\sigma$
  • Reward matches with $+1$

• Resulting score:

$$#\text{matches} - (#\text{mismatches}) \mu - (#\text{indels}) \sigma$$

• Objective: find the best scoring alignment
Number of pairwise alignments

- Given sequences of length $m$ and $n$, the number of alignments is:

$$\sum_{k=0}^{\min(m,n)} \binom{m}{k} \binom{n}{k} = \binom{n+m}{n}$$

- For two sequences of length $n$:

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

Derived using Stirling's approximation: $n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$
Substrings and subsequences

**Definition:** A string $x'$ is a *substring* of a string $x$, if $x = ux'v$ for some prefix string $u$ and suffix string $v$

$$(x' = x_i \ldots x_j, \text{ for some } 1 \leq i \leq j \leq |x|)$$

A string $x'$ is a *subsequence* of a string $x$, if $x'$ can be obtained from $x$ by deleting 0 or more letters

$$(x' = x_{i_1} \ldots x_{i_k}, \text{ for some } 1 \leq i_1 \leq \ldots \leq i_k \leq |x|)$$

**Note:** a substring is always a subsequence

**Example:** $x = \text{abracadabra}$

$y = \text{cadabr};$  
$z = \text{brcdbr};$

**substring**  
**subsequence, not substring**
Encoding alignment as a path in a 2-d grid

<table>
<thead>
<tr>
<th>i coords:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>elements of v</td>
<td>A</td>
<td>T</td>
<td>--</td>
<td>C</td>
<td>--</td>
<td>T</td>
<td>G</td>
<td>A</td>
<td>T</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>elements of w</td>
<td>--</td>
<td>T</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>T</td>
<td>--</td>
<td>A</td>
<td>--</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>j coords:</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

(0,0) → (1,0) → (2,1) → (2,2) → (3,3) → (3,4) → (4,5) → (5,5) → (6,6) → (7,6) → (8,7)

Every alignment is a path in 2-D grid
Alignment as a path
Alignment as a Path in the Edit Graph

\[ \text{ATCGTAC} \]

\[ \text{w} \]

\[ \text{v} \]

- Corresponding path -

\[(0,0) , (1,1) , (2,2) , (2,3) , (3,4) , (4,5) , (5,5) , (6,6) , (7,6) , (7,7)\]
Alignment as a Path in the Edit Graph

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

\[
\begin{array}{cccccccc}
\text{A} & \text{T} & \text{G} & \text{T} & \text{T} & \text{T} & \text{T} & \text{T} \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}
\]

\[\downarrow\]

Represent indels in \(v\) and \(w\) with score -1.

\[\downarrow\]

Represent matches with score 1.

The score of the alignment is 1.
Alignment as a Path in the Edit Graph

Every path in the edit graph corresponds to an alignment:
Alignment algorithms we will cover

- Global alignment
- Local alignment
- Alignment with affine gap penalties
- Scoring matrices
Our simple scoring scheme

- The score when mismatches are penalized by $-\mu$, indels are penalized by $-\sigma$, and matches are rewarded by $+1$:

$$\#\text{matches} - \mu (\#\text{mismatches}) - \sigma (\#\text{indels})$$
Global Alignment: The Needleman-Wunsch algorithm\(^1\)

Find the best alignment between two strings under our scoring scheme

**Input**: Strings \(v\) and \(w\) and a scoring scheme

**Output**: Maximum scoring alignment

\[
s_{i,j} = \max \begin{cases} 
 s_{i-1,j-1} + 1 & \text{if } v_i = w_j \\
 s_{i-1,j-1} - \mu & \text{if } v_i \neq w_j \\
 s_{i-1,j} - \sigma \\
 s_{i,j-1} - \sigma 
\end{cases}
\]

\(s_{i,j}\) – the score for the best alignment of a length \(i\) prefix of \(v\) and a length \(j\) prefix of \(w\)

\(\mu\) : mismatch penalty

\(\sigma\) : indel penalty

\(^1\)A general method applicable to the search for similarities in the amino acid sequence of two proteins, J Mol Biol. 48(3):443–53, 1970.
Needleman Wunsch (cont)

- What about the base case?
NW as a DP algorithm

```python
NW(v,w,sigma,mu):
    for i in range(0, m):
        si,0 = -sigma * i
    for j in range(0, n):
        s0,j = -sigma * j
    for i in range(1, m):
        for j in range(1, n):
            fill in s_i,j
    return (s_m,n)
```

Runtime: \(O(nm)\)
Memory: \(O(nm)\)
Now What?

- The DP algorithm created the alignment grid.
- To read the best alignment: Follow the pointers from sink.
Scoring Matrices

To generalize scoring, we use a **scoring matrix** $\delta$.

Size of the matrix:

- Alignment of DNA sequences: $(4+1) \times (4+1)$
- Alignment of amino acids: $(20+1) \times (20+1)$

The additional row/column includes scores for the gap character “-”

$$ s_{i,j} = \max \begin{cases} 
    s_{i-1,j-1} + \delta(v_i, w_j) \\
    s_{i-1,j} + \delta(v_i, -) \\
    s_{i,j-1} + \delta(-, w_j) 
\end{cases} $$