Local alignment
Ben Langmead

You are free to use these slides. If you do, please sign the guestbook (www.langmead-lab.org/teaching-materials), or email me (ben.langmead@gmail.com) and tell me briefly how you’re using them. For original Keynote files, email me.
Local alignment

Given strings $x$ and $y$, what is the optimal global alignment value of a substring of $x$ to a substring of $y$. This is local alignment.

Assume global alignment scoring where: (a) similarities get $> 0$, (b) dissimilarities get $< 0$, (c) alignment of $\epsilon$ to any string has score $0$

Somehow we must weigh all possible pairs of substrings

What is bound for # substring pairs, assuming $|x| = n$, $|y| = m$? $O(m^2n^2)$
Local alignment

Let $V[i, j]$ be the optimal global alignment value of a substring of $x$ ending at $i$ and a substring of $y$ ending at $j$. The substrings may be empty.

The maximum $V[i, j]$ over all $i, j$ is the optimal score we’re looking for.
Local alignment

How to calculate $V[i,j]$?

Only 4 ways to build a new edit transcript from another one:

- **Vertical**: append $I$ to transcript for $V[i-1,j]$, take gap penalty
- **Horizontal**: append $D$ to transcript for $V[i,j-1]$, take gap penalty
- **Diagonal**: append $M$ or $R$ to transcript for $V[i-1,j-1]$, get match bonus or take replacement penalty as appropriate
- **Empty**: let both substrings be empty, global alignment value = 0

Proof: Gusfield 11.7.1 - 11.7.2
Local alignment

Let $V[0, j] = 0$, and let $V[i, 0] = 0$

Otherwise, let $V[i, j] = \max \begin{cases} V[i - 1, j] + s(x[i - 1], -) \\ V[i, j - 1] + s(-, y[j - 1]) \\ V[i - 1, j - 1] + s(x[i - 1], y[j - 1]) \\ 0 \end{cases}$

$s(a, b)$ assigns a score to a particular match, gap, or replacement

What’s different from global alignment?

First row and columns initialized to all 0s

0 is one of the arguments of the max
Local alignment: Smith-Waterman

Let \( V[0, j] = 0 \), and let \( V[i, 0] = 0 \)

Otherwise, let \( V[i, j] = \max \left\{ \begin{array}{l} V[i - 1, j] + s(x[i - 1], -) \\ V[i, j - 1] + s(-, y[j - 1]) \\ V[i - 1, j - 1] + s(x[i - 1], y[j - 1]) \\ 0 \end{array} \right\} \)

\( s(a, b) \) assigns a score to a particular match, gap, or replacement
Local alignment: Smith-Waterman

Does it make sense that first row and column get all 0s?
Yes, b/c global alignment value of $\epsilon, \epsilon(0)$ always best

$$s(a, b)$$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>C</td>
<td>-4</td>
<td>2</td>
<td>-4</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>G</td>
<td>-4</td>
<td>-4</td>
<td>2</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>T</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>2</td>
<td>-6</td>
</tr>
<tr>
<td>-</td>
<td>-6</td>
<td>-6</td>
<td>-6</td>
<td>-6</td>
<td>-6</td>
</tr>
</tbody>
</table>
Local alignment: Smith-Waterman

\[ V[i, j] = \max \begin{cases} 
V[i - 1, j] + s(x[\ell - 1], -) \\
V[i, j - 1] + s(-, y[j - 1]) \\
V[i - 1, j - 1] + s(x[\ell - 1], y[j - 1]) \\
0 
\end{cases} \]
Local alignment: Smith-Waterman

\[
V[i, j] = \max \left\{ \begin{array}{l}
V[i - 1, j] + s(x[i - 1], -) \\
V[i, j - 1] + s(-, y[j - 1]) \\
V[i - 1, j - 1] + s(x[i - 1], y[j - 1]) \\
0
\end{array} \right\}
\]

\[
\begin{array}{cccccccccccc}
\epsilon & T & A & T & A & T & G & C & G & C & G & T & T & T \\
\hline
\epsilon & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 2 & 0 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 4 & 0 & 2 & 0 \\
T & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\
A & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
T & 0 & 2 & 0 & 6 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
G & 0 & 0 & 0 & 0 & 2 & 0 & 8 & 2 & 2 & 2 & 0 & 2 & 0 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 10 & 4 & 0 & 4 & 0 & 0 \\
T & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 4 & 6 & 0 & 0 & 0 & 2 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 6 & 8 & 2 & 2 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 8 & 4 & 4 & 0 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 2 & 10 & 4 & 0 & 0 \\
G & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 6 & 2 & 4 & 12 & 6 \\
C & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 2 & 4 & 6 & 8 & 2 \\
T & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \\
A & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 4
\end{array}
\]

\[
s(a, b)
\]

\[
\begin{array}{cccccc}
A & C & G & T & - \\
\hline
A & 2 & -4 & -4 & -4 & -6 \\
C & -4 & 2 & -4 & -4 & -6 \\
G & -4 & -4 & 2 & -4 & -6 \\
T & -4 & -4 & -4 & 2 & -6 \\
- & -6 & -6 & -6 & -6 & -6
\end{array}
\]

0’s in essence allow peaks of similarity to rise above “background” of 0s
Local alignment: Smith-Waterman

Backtrace: (a) start from *maximal* cell in the matrix, (b) stop backtrace when we reach a cell with score = 0

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>G</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th>T</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</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>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</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>4</td>
<td>0</td>
<td>1</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>2</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>5</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>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>2</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>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ s(a, b) \]

- A 2 -4 -4 -4 -6
- C -4 2 -4 -4 -6
- G -4 -4 2 -4 -6
- T -4 -4 -4 -2 -6
- - -6 -6 -6 -6

\[ y : \quad T A T A T G C - G G C G T T T \]
\[ x : \quad G G T A T G C T G G C G C T A \]
Local alignment: Smith-Waterman

What if we didn’t have a positive “bonus” for matches?

All cells would = 0

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>C</td>
<td>-4</td>
<td>2</td>
<td>-4</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>G</td>
<td>-4</td>
<td>-4</td>
<td>2</td>
<td>-4</td>
<td>-6</td>
</tr>
<tr>
<td>T</td>
<td>-4</td>
<td>-4</td>
<td>-4</td>
<td>2</td>
<td>-6</td>
</tr>
<tr>
<td>-</td>
<td>-6</td>
<td>-6</td>
<td>-6</td>
<td>-6</td>
<td></td>
</tr>
</tbody>
</table>

What if we didn’t have negative “penalties” for edits?

Rule for $\epsilon$, $\epsilon$ would never be used and alignment would essentially be global

$$s(a, b) = \max \left\{ V[i-1, j] + s(x[i-1], -) \right\}$$

What if we didn’t have negative “penalties” for edits?

Rule for $\epsilon$, $\epsilon$ would never be used and alignment would essentially be global

$$s(a, b) = \max \left\{ V[i-1, j] + s(-, y[j-1]) \right\}$$

$$V[i-1, j-1] + s(x[i-1], y[j-1])$$

$$0$$
Local alignment: Smith-Waterman

```python
def smithWaterman(x, y, s):
    """ Calculate local alignment values of sequences x and y using dynamic programming. Return maximal local alignment value. """
    V = numpy.zeros((len(x)+1, len(y)+1), dtype=int)
    for i in xrange(1, len(x)+1):
        for j in xrange(1, len(y)+1):
            V[i, j] = max(V[i-1, j-1] + s(x[i-1], y[j-1]),  # diagonal
                           V[i-1, j ] + s(x[i-1], '-'),  # vertical
                           V[i , j-1] + s('-', y[j-1]),  # horizontal
                           0)                           # empty
    argmax = numpy.where(V == V.max())
    return int(V[argmax])
```

Python example: [http://nbviewer.ipython.org/6994170](http://nbviewer.ipython.org/6994170)
Local alignment: Smith-Waterman

We might be interested in the best local alignment, or in many good-enough local alignments

Reducing good-enough threshold risks allowing lots of tiny alignments that aren’t very relevant