Alignments 2: Local alignment

Sequence Analysis
06-11-2006

What can sequence alignment tell us about structure
HSSP  Sander & Schneider, 1991

Pair-wise alignment
Complexity of the problem

Combinatorial explosion
- 1 gap in 1 sequence: \( n+1 \) possibilities
- 2 gaps in 1 sequence: \( (n+1)n \)
- 3 gaps in 1 sequence: \( (n+1)n(n-1) \), etc.

\[
\begin{align*}
2n & = \frac{(2n)!}{n!^2} \sim \frac{2^n}{\sqrt{n\pi}} \\
2 \text{ sequences of } 300 \text{ a.a.: } & \sim 10^{88} \text{ alignments} \\
2 \text{ sequences of } 1000 \text{ a.a.: } & \sim 10^{600} \text{ alignments!}
\end{align*}
\]

The algorithm for linear gap penalties

Scoring alignments

- Substitution (or match/mismatch)
  - DNA
  - proteins
- Gap penalty
  - Linear: \( gp(k) = \alpha k \)
  - Affine: \( gp(k) = \beta + \alpha k \)
  - Concave, e.g.: \( gp(k) = \log(k) \)

The score for an alignment is the sum of the scores over all alignment columns

General alignment score:

\[
S_{i,j} = \sum \alpha(i,h) - \sum N \cdot gp(0)
\]

Scoring and gap penalties

- Scoring:
  - DNA or protein?
  - Which evolutionary distance do we cover?
- Gap penalty: always less than 0 (i.e. they lower the alignment score)
  - Linear: \( g(k) = \alpha k \)
  - Affine: \( g(k) = \beta + \alpha k \) ← harder to implement
  - Concave: \( g(k) = \log(k) \) ← no efficient solution
Global dynamic programming – general algorithm

\[ M[i,j] = \max \{ M[i-1,j-1] + \text{score}(X[i],Y[j]),
\max \{ M[i-1,j] - g, \text{open gap penalty} \},
\max \{ M[i,j-1] - g, \text{extension penalty} \} \} \]

This more general way of dynamic programming also allows for affine or other gap-penalty regimes.

Value from residue exchange matrix

Number of gap-extensions

Gap-open penalty

Gap-extension penalty

Note about gap penalties

- Some affine schemes use
  \[ \text{gap\_penalty} = -g \times |i-1|, \]
  while others use
  \[ \text{gap\_penalty} = -g \times \text{extension} \times l, \]
  where \( l \) is the length of the gap.

One can be converted into the other by adapting the \( g_{\text{open}} \) penalty.

Variation on global alignment

- **Global alignment**: the previous algorithm is called global alignment, because it uses all letters from both sequences.

  \[
  \text{CAGCACTTGGATTCTCGG}
  \]

- **Semi-global alignment**: don’t penalize for start/end gaps (omit the start/end of sequences).

  \[
  \text{CAGCACTTGGATTCTCGG}
  \]

  - Applications of semi-global:
    - Finding a gene in genome
    - Placing marker onto a chromosome
  - One sequence much longer than the other
  - Danger – really bad alignments for divergent seqs

Easy DP recipe for using affine gap penalties

- \( M[i,j] \) is optimal alignment (highest scoring alignment until \([i,j]\))
- Check
  - \( \text{Cell}[i-1,j-1] \): apply score for \([i-1,j-1]\)
  - preceding row until \([i-2,j]\): apply appropriate gap penalties
  - preceding column until \([i,j-1]\): apply appropriate gap penalties

Global dynamic programming

- \( g_{\text{open}} = 10 \), \( g_{\text{extension}} = 2 \)

Global alignments - review

- Take two sequences: \( X[i] \) and \( Y[j] \)
- The best alignment for \( X[1..i] \) and \( Y[1..j] \) is called \( M[i,j] \)
- Initiation: \( M[0,0] = 0 \)
- Apply the equation
- Find the alignment with backtracking
Global and local alignment

Local alignment

• What’s local?
  – Allow only parts of the sequence to match
  – Results in High Scoring Segments
  – Locally maximal: cannot make it better by trimming/extending the alignment

Domains - example
Immunoglobulin domain

Global → local alignment

a) global align

b) retrieve the result

c) sum score along the result

Local alignment – breaking the alignment

• A recipe
  – Just don’t let the score go below
  – Start the new alignment when it happens
  – Where is the result in the matrix?

Before:

After:
Local alignment – the equation

\[
M[i, j] = \max (M[i-1, j-1] + \text{score}(X_i, Y_j), M[i-1, j] - g, M[i, j-1] - g)
\]

- Init the matrix with 0’s
- Read the maximal value from anywhere in the matrix
- Find the result with backtracking

Finding second best alignment

- We can find the best local alignment in the sequence
- But where is the second best one?

Example: repeats proteins

Local alignment traces showing similarity between repeats

Note: repeats are similar motifs but need not be identical

Finding second best alignment

- Don’t let any matched pair contribute to the next alignment

Clumps gene X

The figure shows two proteins with so-called tandem repeats (similar motifs adjacent in the sequence)

'Shadows' of various alignments are visible – these all derive from parts of a same locally optimal alignment
Extraction of alignments – Waterman-Eggert algorithm

1. Repeat
   a. Retrieve the highest scoring alignment
   b. Set its trace to 0

Example: general algorithm for local alignment

DP algorithm with affine gap penalties (PAM250, Po=10, Pe=2)

<table>
<thead>
<tr>
<th>seq X:</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>seq Y:</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

Extra start/end columns not necessary (no end-gaps). Each negative scoring cell is set to zero. Highest scoring cell may be found anywhere in search matrix after calculating it. Trace highest scoring cell back to first cell with zero value (or the beginning of one or both sequences).

Pitfalls of alignments

- Alignment is often not a reconstruction of evolution (common ancestor is usually extinct by the time of alignment)
- Repeats: matches to the same fragment
### Summary

1. **Global**
   - e.g. Needleman-Wunsch algorithm
2. **Semi-global**
3. **Local**
   - e.g. Smith-Waterman algorithm
4. **Many local alignments**
   - aka Waterman-Eggert algorithm

What’s the number of steps in these algorithms?

How much memory is used?

---

### Semi-global alignment

- **Ignore start gaps**
  - First row/column set to 0
- **Ignore end gaps**
  - Read the result from last row/column

---

### Low complexity regions

- Local composition bias
  - Replication slippage: e.g. triplet repeats
- Results in spurious hits
  - Breaks down statistical models
  - Different proteins reported as hits due to similar composition
  - Up to ¼ of the sequence can be biased
- Widely used filtering program: SEGS (Wootton and Federhen, 1996)

---

### Synteny

- **Synteny is preservation of gene order**
  - 5 to 20 million years of divergence

Figure shows preservation of gene order in four yeast species

Arrows in figure represent genes; arrow direction indicates ‘+’ or ‘-’ strand of DNA

Sequencing and comparison of yeast species to identify genes and regulatory elements


---

### Take-home messages (I)

- Global, semi-global and local alignment
- Three types of gap penalties
- ‘easy’ (fast) and general algorithm
- Pitfalls of local alignment
- Low-complexity regions
Take-home messages (II)

Make sure you understand and can carry out
1. the ‘simple’ DP algorithm (for linear gap penalties)
2. The general DP algorithm for global, semi-global and local alignment!