Alignments 2
Local alignments

Sequence Analysis
Lecture 3

06-12-2007

Pair-wise alignment
Complexity of the problem

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

$$2^n \cdot \frac{(2n)!}{n!} \approx \frac{2^n}{\sqrt{n}}$$

2 sequences of 300 a.a.: $\approx 10^{20}$ alignments
2 sequences of 1000 a.a.: $\approx 10^{60}$ alignments!

The algorithm for linear gap penalties

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

Dynamic programming
Scoring alignments

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

General alignment score:

$$S_{ab} = \sum x_{a,b} - \sum N \cdot g_p(k)$$
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: $g(k)=ak$
  - Affine: $g(k)=b+ak$
  - Concave: $g(k)=\log(k)$

Variation on global alignment

Global dynamic programming – general algorithm

Note about gap penalties

- Some affine schemes use
  $$ \text{gap\_penalty} = \text{gap\_open} - \text{extension}\_l$$
  while others use
  $$ \text{gap\_penalty} = \text{gap\_open} - \text{extension}\_l$$
  where $l$ is the length of the gap.

One can be converted into the other by adapting the $\text{gap\_open}$ penalty.

Global dynamic programming

Variation on global alignment

- **Global** alignment: the previous algorithm is called global alignment, because it uses all letters from both sequences.
  - **Semi-global** alignment: don’t penalize for start/end gaps (omit the start/end of sequences).

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

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

Global and local alignment

- \( X \) and \( Y \) sequences
- \( A \) and \( C \) matched subsequeces
- \( B \) local mismatch

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

Why local?

- Parts of sequence diverge faster
- Evolutionary pressure does not put constraints on the whole sequence
- Proteins have modular construction

Global → local alignment

- Take the old, good equation
- Look at the result of the global alignment

Domains - example

Immunoglobulin domain
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

\[ \begin{align*}
M[i, j] &= \max \left( 
M[i-1, j-1] + \text{score}(X[i], Y[j]), 
M[i-1, j], 
M[i, j-1] - g
\right) 
\end{align*} \]

- Init the matrix with 0’s
- Fill in the search matrix (forward step)
- Read the maximal value from anywhere in the matrix
- Find the result by performing trace-back

Finding second best alignment

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

Scoring:
1 for match
-2 for a gap

Best alignment

A clump

Clump of an alignment

- Alignments sharing at least one aligned pair

Example: repeats proteins

Local alignment traces showing similarity between repeats

Note: repeats are similar motifs but need not be identical

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 optimised alignment
Finding second best alignment

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

Clumps

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

The highest scoring local alignment is indicated

Clumps

gene X

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)

Extra start/end columns/rows 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)

General local DP algorithm for using affine gap penalties

\[
\begin{align*}
\text{Penalty} &= P_{\text{open}} + \text{gap}\_\text{length} \times P_{\text{extension}} \\
\mathbf{M}[i][j] &= \begin{cases} 
\max(i-1, j-1) & \text{if } i > 0, j > 0, \text{ no gap penalties} \\
\min(i-1, j) & \text{if } i > 0, j = 0, \text{ no gap penalties} \\
\min(i, j-1) & \text{if } i = 0, j > 0, \text{ no gap penalties} \\
\max(i-1, j-1) - \text{gap penalty} & \text{if } i > 0, j > 0, \text{ gap penalties} \\
\min(i-1, j) - \text{gap penalty} & \text{if } i > 0, j = 0, \text{ gap penalties} \\
\min(i, j-1) - \text{gap penalty} & \text{if } i = 0, j > 0, \text{ gap penalties} \\
\end{cases}
\end{align*}
\]
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

Global alignment (“simple” recursive DP)

- Fill pre-row and pre-column with gap penalties to account for initial end gaps
- Perform trace-back from lower-right matrix cell, thereby accounting for terminal end gaps

Semi-global alignment

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

Local alignment

- Ignore cells that score <= 0
- Read the result (local alignment score) from the highest cell in the matrix
  - Perform trace-back starting from this cell.

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

n. What’s the number of steps in these algorithms?
n. How much memory is used?

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 5% of the sequence can be biased
- Widely used filtering program: SEGS (Wootton and Federhen, 1996)

Huntington’s disease

- Huntington gene of unknown function
- Repeats 8-55 normal, 26-120 disease

Alignment is often not a reconstruction of evolution (common ancestor is usually extinct by the time of alignment). Repeats: matches to the same fragment. Global alignment (“simple” recursive DP): Fill pre-row and pre-column with gap penalties to account for initial end gaps. Perform trace-back from lower-right matrix cell, thereby accounting for terminal end gaps. Semi-global alignment: Ignore initial end gaps (first row/column set to 0) and terminal end gaps (read the result from last row/column). Local alignment: Ignore cells that score <= 0; read the result (local alignment score) from the highest cell in the matrix and perform trace-back starting from this cell. Summary: Global, e.g. Needleman-Wunsch algorithm; Semi-global; Local, e.g. Smith-Waterman algorithm; Many local alignments aka Waterman-Eggert algorithm. How many steps in these algorithms? How much memory is used? Low complexity regions: Local composition bias due to replication slippage (e.g. triplet repeats), results in spurious hits due to breakdown of statistical models, different proteins reported as hits due to similar composition, up to 5% of the sequence can be biased. Widely used filtering program: SEGS (Wootton and Federhen, 1996). Huntington’s disease involves the Huntington gene of unknown function with repeats ranging from 8-55 normal to 26-120 disease.
**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)**

- Know global, semi-global and local alignment (three types)
- Know three types of gap penalties
- ‘easy’ (fast) and general algorithm
- Pitfalls of global, semi-global and 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, but with extension to affine penalties) for global, semi-global and local alignment

2. The general DP algorithm for global, semi-global and local alignment!