Pairwise Alignment and Database Searching

Anders Gorm Pedersen
Henrik Nielsen
Center for Biological Sequence Analysis

Sequences are related

- Darwin: all organisms are related through descent with modification
- => Sequences are related through descent with modification
- => Similar molecules have similar functions in different organisms

Phylogenetic tree based on ribosomal RNA: three domains of life

Sequences are related, II

Phylogenetic tree of globin-type proteins found in humans
Why compare sequences?

- Determination of evolutionary relationships
- Prediction of protein function and structure (database searches).

Protein 1: binds oxygen

Sequence similarity

Protein 2: binds oxygen?

Dotplots: visual sequence comparison

1. Place two sequences along axes of plot
2. Place dot at grid points where two sequences have identical residues
3. Diagonals correspond to conserved regions

Pairwise alignments

<table>
<thead>
<tr>
<th></th>
<th>Global alignment score: 374</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>43.2% identity:</td>
</tr>
<tr>
<td></td>
<td>10  20  30  40  50</td>
</tr>
<tr>
<td>alpha</td>
<td>P-LPQGNVYHREYFPEDDLEELLHRPN-CL---GQEE</td>
</tr>
<tr>
<td>beta</td>
<td>VCEPPQGPQAEVNIYAVVAGFEGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA</td>
</tr>
<tr>
<td></td>
<td>10  20  30  40  50  60  70  80  90</td>
</tr>
<tr>
<td></td>
<td>100 110</td>
</tr>
<tr>
<td>alpha</td>
<td>QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL</td>
</tr>
<tr>
<td>beta</td>
<td>KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF</td>
</tr>
<tr>
<td></td>
<td>60  70  80  90  100  110</td>
</tr>
<tr>
<td>alpha</td>
<td>PAEFTPAVHASLDKFLASVSTVLTSKYR</td>
</tr>
<tr>
<td>beta</td>
<td>GKEFTPPVQAAYQKVVAGVANALAHKYH</td>
</tr>
<tr>
<td></td>
<td>120 130 140</td>
</tr>
</tbody>
</table>
Pairwise alignment

100.000% identity in 3 aa overlap

SPA
:::
SPA

Percent identity is not a good measure of alignment quality

Pairwise alignments: alignment score

43.2% identity:  Global alignment score: 374

alpha  V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS-----HGSA
beta   VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNP

Alignment scores: match vs. mismatch

Simple scoring scheme (too simple in fact…):

Matching amino acids: 5
Mismatch: 0

Scoring example:

<table>
<thead>
<tr>
<th></th>
<th>K</th>
<th>A</th>
<th>W</th>
<th>S</th>
<th>A</th>
<th>D</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5+5+5+5+5 = 25
**Pairwise alignments: conservative substitutions**

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
</table>

**Amino acid properties**

- Serine (S) and Threonine (T) have similar physicochemical properties
- Aspartic acid (D) and Glutamic acid (E) have similar properties

**Protein substitution matrices**

| A | 5 |
| B | -1 | 7 |
| C | -1 | -1 | 1 |
| D | 0 | 2 | 0 |
| E | -1 | -1 | -1 | 2 |
| F | 0 | 0 | 3 | 2 |
| G | 0 | -1 | -1 | -1 | 2 |
| H | -1 | 1 | 0 | 0 | 2 |
| I | -1 | 0 | 1 | 0 | 0 |
| K | -1 | 0 | 1 | 0 | 0 |
| L | 0 | 1 | 0 | 1 | 0 |
| M | 0 | 1 | 0 | 1 | 0 |
| N | 0 | 1 | 0 | 1 | 0 |
| O | 0 | 1 | 0 | 1 | 0 |
| P | 0 | 1 | 0 | 1 | 0 |
| Q | 0 | 1 | 0 | 1 | 0 |
| R | 0 | 1 | 0 | 1 | 0 |
| S | 0 | 1 | 0 | 1 | 0 |
| T | 0 | 1 | 0 | 1 | 0 |
| U | 0 | 1 | 0 | 1 | 0 |
| V | 0 | 1 | 0 | 1 | 0 |

**RLOCUSM0 matrix:**

- Positive scores on diagonal (identities)
- Similar residues get higher (positive) scores
- Dissimilar residues get smaller (negative) scores
Pairwise alignments: insertions/deletions

<table>
<thead>
<tr>
<th>Pairwise alignments: insertions/deletions</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.2% identity; Global alignment score: 374</td>
</tr>
</tbody>
</table>

| alpha | V-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLS | HGSA |
| beta  | VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLS | TPDAV MGNP |

| alpha | QVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHL | .::.:::::  :.....::.:.. .....::.::  ::.::: ::.::.. :. .:: :. |
| beta  | KVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHF |

| alpha | PAEFTPAVHASLDKFLASVSTVLTSKYR |
| beta  | GKEFTPPVQAAYQKVVAGVANALAHKYH |

Affine gap penalties:
Multiple insertions/deletions may be one evolutionary event => Separate penalties for gap opening and gap elongation

Handout

Compute 4 alignment scores: two different alignments using two different alignment matrices (and the same gap penalty system)

Score 1: Alignment 1 + BLOSUM-50 matrix + gaps
Score 2: Alignment 1 + BLOSUM-Trp matrix + gaps
Score 3: Alignment 2 + BLOSUM-50 matrix + gaps
Score 4: Alignment 2 + BLOSUM-Trp matrix + gaps

Note: fake matrix constructed for pedagogic purposes.
Handout: summary of results

<table>
<thead>
<tr>
<th>Alignment 1</th>
<th>Alignment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOSUM-50</td>
<td></td>
</tr>
<tr>
<td>BLOSUM-Trp</td>
<td></td>
</tr>
</tbody>
</table>

Protein substitution matrices

BLOSUM50 matrix:
- Positive scores on diagonal (identities)
- Similar residues get higher (positive) scores
- Dissimilar residues get smaller (negative) scores

Protein substitution matrices: different types

- **Identity matrix**
  (match vs. mismatch)

- **Genetic code matrix**
  (how similar are the codons?)

- **Chemical properties matrix**
  (use knowledge of physicochemical properties to design matrix)

- **Empirical matrices**
  (based on observed pair-frequencies in hand-made alignments)
  - PAM series
  - BLOSUM series
  - Gonnet
Estimation of an empirical matrix

- Start from given alignments of closely related proteins
- Count the aligned amino acid pairs (e.g., A aligned with A makes up 1.5% of all pairs. A aligned with C makes up 0.01% of all pairs, etc.)
- Expected pair frequencies are computed from single amino acid frequencies. (e.g., $f_{A,C} = f_A \times f_C = 7\% \times 3\% = 0.21\%$).
- For each amino acid pair the substitution scores are essentially computed as:
  $$S_{AC} = \log \frac{\text{Pair-freq(obs)}}{\text{Pair-freq(expected)}}$$

To obtain the PAM1 matrix, normalize pair frequencies to 1% difference before applying the logarithm
To obtain the PAM50 matrix, extrapolate the PAM1 matrix via matrix multiplication

---

Estimation of the BLOSUM 50 matrix

- Use the BLOCKS database (ungapped alignments of especially conserved regions of multiple alignments)
- For each alignment in the BLOCKS database the sequences are grouped into clusters with at least 50% identical residues (for BLOSUM 50)
- All pairs of sequences are compared between clusters, and the observed pair frequencies are noted
- Substitution scores are calculated as before

---

Pairwise alignment

Optimal alignment:

alignment having the highest possible score given a substitution matrix and a set of gap penalties

So:

best alignment can be found by exhaustively searching all possible alignments, scoring each of them and choosing the one with the highest score?
The problem: How many possible alignments are there?

Pairwise alignment: the problem

The number of possible pairwise alignments increases explosively with the length of the sequences:

\[ f(n_1, n_2) = \sum_{i=0}^{n_1} \binom{n_1}{i} (n_2 + i)^{n_1 - i}. \]

Two protein sequences of length 100 amino acids can be aligned in approximately \(10^{60}\) different ways.

Time needed to test all possibilities is same order of magnitude as the entire lifetime of the universe.

Pairwise alignment: the solution

“Dynamic programming” (the Needleman-Wunsch algorithm)
Alignment depicted as path in matrix

Meaning of point in matrix: all residues up to this point have been aligned (but there are many different possible paths).

Position labeled "x": TC aligned with TC

Any given point in matrix can only be reached from three possible previous positions (you cannot "align backwards").

Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.
Dynamic programming: computation of scores

Any given point in matrix can only be reached from three possible positions (you cannot "align backwards").

- Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

\[
score(x, y) = \max \left\{ \begin{array}{c}
score(x-1, y-1) - \text{gap penalty} \\
(score(x-1, y-1) + \text{substitution-score}(x, y)) \\
(score(x-1, y) - \text{gap penalty})
\end{array} \right. \]

Dynamic programming: computation of scores

Any given point in matrix can only be reached from three possible positions (you cannot "align backwards").

- Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

\[
score(x, y) = \max \left\{ \begin{array}{c}
score(x-1, y-1) - \text{gap penalty} \\
(score(x-1, y-1) + \text{substitution-score}(x, y)) \\
(score(x-1, y) - \text{gap penalty})
\end{array} \right. \]
Dynamic programming: computation of scores

Any given point in matrix can only be reached from three possible positions (you cannot ‘align backwards’).

Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

Each new score is found by choosing the maximum of three possibilities. For each square in matrix, keep track of where best score came from.

Fill in scores one row at a time, starting in upper left corner of matrix, ending in lower right corner.

\[
score(x,y) = \max \begin{cases} 
score(x,y-1) \cdot \text{gap-penalty} \\
score(x-1,y) + \text{substitution-score}(x,y) \\
score(x-1,y-1) \cdot \text{gap-penalty}
\end{cases}
\]

Dynamic programming: example

Dynamic programming: example

Dynamic programming: example
Global versus local alignments

Global alignment: align full length of both sequences.
(The "Needleman-Wunsch" algorithm).

Local alignment: find best partial alignment of two sequences
(the "Smith-Waterman" algorithm).

Local alignment overview

- The recursive formula is changed by adding a fourth possibility: zero. This means local alignment scores are never negative.

\[
\text{score}(x,y) = \max\begin{cases} 
\text{score}(x-1,y) - \text{gap-penalty} \\
\text{score}(x,y-1) - \text{gap-penalty} \\
\text{score}(x-1,y-1) + \text{substitution-score}(x,y) \\
0
\end{cases}
\]

- Trace-back is started at the highest value rather than in lower right corner
- Trace-back is stopped as soon as a zero is encountered

Local alignment: example
Substitution matrices and sequence similarity

Substitution matrices come as series of matrices calculated for different degrees of sequence similarity (different evolutionary distances).

<table>
<thead>
<tr>
<th>&quot;Hard&quot; matrices</th>
<th>&quot;Soft&quot; matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designed for very similar sequences</td>
<td>Designed for less similar sequences</td>
</tr>
<tr>
<td>High numbers in the BLOSUM series (e.g., BLOSUM90)</td>
<td>Low numbers in the BLOSUM series (e.g., BLOSUM30)</td>
</tr>
<tr>
<td>Low numbers in the PAM series (e.g., PAM30)</td>
<td>High numbers in the PAM series (e.g., PAM250)</td>
</tr>
<tr>
<td>Severe mismatch penalties</td>
<td>Less severe mismatch penalties</td>
</tr>
<tr>
<td>Yield short alignments with high %identity</td>
<td>Yield longer alignments with lower %identity</td>
</tr>
</tbody>
</table>

Alignments: things to keep in mind

"Optimal alignment" means "having the highest possible score, given substitution matrix and set of gap penalties". This is NOT necessarily the biologically most meaningful alignment.

Specifically, the underlying assumptions are often wrong: substitutions are not equally frequent at all positions, affine gap penalties do not model insertion/deletion well, etc.

Pairwise alignment programs always produce an alignment - even when it does not make sense to align sequences.