# CISC 889 Bioinformatics (Spring 2004)

## Sequence pairwise alignment (I)



## Contents

Alignment algorithms

- Needleman-Wunsch (global alignment)
- Smith-Waterman (local alignment)
- Heuristic algorithms
  - FASTA
  - BLAST

## Sequence Alignment

- Motivation
  - Sequence assembly: reconstructing long DNA sequences from overlapping sequence fragments
  - Annotation: assign functions to newly discovered genes
    - Raw genomic (DNA) sequences -> coding sequences (CDS), candidate for genes -> protein sequence -> function
    - Terminologies: cDNA, RNA, mRNA
    - Evolution: mutation -> sequence diversity (vs homology) -> (new) phenotype ?
    - Corner stone: sequence similarity -> sequence homology -> same function



## Alignment algorithms

- What is an alignment?

  A one-to-one matching of two sequences so that each character in a pair of sequences is associated with a single character of the other sequence or with a null character (gap). Alignments are often displayed as two rows with an optional third row in between pointing out regions of similarity.
- Example:

```
>gi|7434520|pir||G64632 acetate kinase - Helicobacter pylori (strain 26695)
          Length = 388

 Score = 35.8 bits (81), Expect = 0.10
 Identities = 21/51 (41%), Positives = 29/51 (56%), Gaps = 2/51 (3%)

Query: 1  VLVLNCGSSSLKFAIIDAVNGEEYLSGLAECF--HLPEARIKWKMDGNKQE 49
          +LVLN GSSS+KF + D    +   SGLAE     + + +IK  +  N QE
Sbjct: 3  ILVLNLGSSSIKFKLFDMKENKPLASGLAEKIGEEIGQLKIKSHLHHNDQE 53
```

- Types of alignment:
  - pairwise vs multiple;
  - global vs local
- Algorithms
  - Rigorous
  - heuristic

## PAM matrices (Margaret Dayhoff, 1978)

- point accepted mutation or percent accepted mutation
- unit of measurement of evolutionary divergence between two amino acid sequences
- substitute matrices (scoring matrices)

1 PAM = one accepted point-mutation event per one-hundred amino acids



## PAM (cont'd)

caveat:

- Sequences s1 and s2 are x PAM divergent does not imply s1 and s2 have x percent sequence difference ( should be equal or less).

facts:

1. even amino acid sequences that have diverged by 200 PAM units are expected to be identical in about 25% of their positions.
2. sequences that are 250 PAM units diverged can generally be distinguished from a pair of random sequences.

PAM matrix is a 20 by 20 matrix, and each element $p_{ij}$ represents the expected evolutionary exchange between the two corresponding amino acids for sequences that are a specific number of PAM units diverged. That is,

$$p_{ij} = \log[f(i,j)/f(i)f(j)]$$

where f(i) and f(j) are the frequencies that amino acids $A_i$ and $A_j$ appear in the sequences, and f(i,j) the frequency that $A_i$ and $A_j$ are aligned.

By construction, PAM n matrices with n > 1 are extrapolated from those with lower n. Namely, it assumes that the frequencies of the amino acids remain constant over time and that the mutational processes causing replacements in an interval of one PAM unit operate the same for longer periods.



Ideally, to align two sequences that are x PAM units diverged, the PAM x matrix should be used.

Practically, since you do not know *a priori* how much diverged given two sequences, you may need to try different PAM matrices.

Empirically, PAM 250 matrix is reported to be very effective for aligning sequences used in evolutionary studies.

PAM 250

```
C  12
S   0  2
T  -2  1  3
P  -3  1  0  6
A  -2  1  1  1  2
G  -3  1  0 -1  1  5
N  -4  1  0 -1  0  0  2
D  -5  0  0 -1  0  1  2  4
E  -5  0  0 -1  0  0  1  3  4
Q  -5 -1 -1  0  0 -1  1  2  2  4
H  -3 -1 -1  0 -1 -2  2  1  1  3  6
R  -4  0 -1  0 -2 -3  0 -1 -1  1  2  6
K  -5  0  0 -1 -1 -2  1  0  0  1  0  3  5
M  -5 -2 -1 -2 -1 -3 -2 -3 -2 -1 -2  0  0  6
I  -2 -1  0 -2 -1 -3 -2 -2 -2 -2 -2 -2 -2  2  5
L  -6 -3 -2 -3 -2 -4 -3 -4 -3 -2 -2 -3 -3  4  2  6
V  -2 -1  0 -1  0 -1 -2 -2 -2 -2 -2 -2 -2  2  4  2  4
F  -4 -3 -3 -5 -4 -5 -4 -6 -5 -5 -2 -4 -5  0  1  2 -1  9
Y   0 -3 -3 -5 -3 -5 -2 -4 -4 -4  0 -4 -4 -2 -1 -1 -2  7 10
W  -8 -2 -5 -6 -6 -7 -4 -7 -7 -5 -3  2 -3 -4 -5 -2 -6  0  0 17
-----------------------------------------------------------------
   C  S  T  P  A  G  N  D  E  Q  H  R  K  M  I  L  V  F  Y  W
```



BLOSUM matrices [Steven and Jorja Henikoff]

- BLOSUM x matrix is a 20 by 20 matrix. Its elements are defined like those of PAM matrices but the frequencies are collected from sequences in BLOCKs database that are less than x percent identical (generally x is between 50 and 80).

- By their construction, BLOSUM matrices are believed to be more effectively detect distant homology.

- in place of PAM 250, BLOSUM 62 is the default matrix used in database search.

PROSITE is a "dictionary of sites and patterns in proteins" that is linked to the protein sequence database Swiss-Prot.

The signature patterns in PROSITE are written as regular expressions.

For example, G-[GN]-[SGA]-G-x-R-[SGA]-C-x(2)-[IV] is a PROSITE pattern.

BLOCKS is a database of protein motifs that is derived from the PROSITE library.



Needleman-Wunsch algorithm (Global Pairwise optimal alignment, 1970)

To align two sequences x[1...n] and y[1...m],

i) if x at i aligns with y at j, a score $s(x_i, y_j)$ is added; if either $x_i$ or $y_j$ is a gap, a score of d is subtracted (penalty).

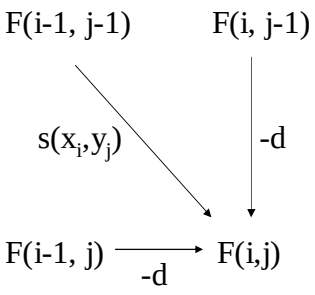
ii) The best score up to (i,j) will be

$$F(i,j) = \max \{ F(i-1, j-1) + s(x_i, y_j),$$
$$F(i-1,j) - d,$$
$$F(i, j-1) - d$$
$$\}$$

Needleman-Wunsch (cont'd)

iii) Tabular computing to get F(i,j) for all $1<i<n$ and $i<j<m$

Draw a diagram:




By definition, F(n,m) gives the best score for an alignment of x[1...n] and y[1...m].



iv) Trace-back

To find the alignment itself, we must find the path of choices (in applying the formulae of ii) when tabular computing that led to this final value.

> Vertical move is gap in the column sequence.
> Horizontal move is gap in the row sequence.
> Diagonal move is a match.

Example: Align HEAGAWGHEE and PAWHEAE.

Use BLOSUM 50 for substitution matrix and d=-8 for gap penalty.

(page 21 in the text)

| | | H | E | A | G | A | W | G | H | E | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **0** | **-8** | **-16** | -24 | -32 | -40 | -48 | -56 | -64 | -72 | -80 |
| **P** | -8 | -2 | -9 | **-17** | **-25** | -33 | -42 | -49 | -57 | -65 | -73 |
| **A** | -16 | -10 | -3 | -4 | -12 | **-20** | -28 | -36 | -44 | -52 | -60 |
| **W** | -24 | -18 | -11 | -6 | -7 | -15 | **-5** | **-13** | -21 | -29 | -37 |
| **H** | -32 | -14 | -18 | -13 | -8 | -9 | -13 | -7 | **-3** | -11 | -19 |
| **E** | -40 | -22 | -8 | -16 | -16 | -9 | -12 | -15 | -7 | **3** | -5 |
| **A** | -48 | -30 | -16 | -3 | -11 | -11 | -12 | -12 | -15 | **-5** | 2 |
| **E** | -56 | -38 | -24 | -11 | -6 | -12 | -14 | -15 | -12 | -9 | **1** |



Time complexity: O(nm)

Space complexity: O(nm)

Big-O notation:

f(x) = O(g(x)) => f is upper bound by g

f(x) = Omega(g(x)) => f is lower bound by g

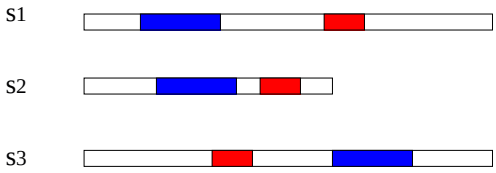
f(x) = Theta(g(x)) => f is bound to g within constant factors

Local pairwise optimal alignment

why need local alignment (vs global)?

- mosaic structure ( functioning domains) of proteins

e.g., are these three sequences similar or not?






# Local alignment

- naive algorithm:
  - there are $\Theta(n^2 m^2)$ pairs of substrings; to align each pair as a global alignment problem will take $O(nm)$; the optimal local alignment will therefore take $O(n^3 m^3)$.
- Smith-Waterman algorithm (dynamic programming)
  recurrence relationship

$$F(i,j) = \max \{ 0,\ F(i-1, j-1) + s(x_i, y_j),\ F(i-1,j) - d,\ F(i, j-1) - d \}$$

Notes: 1) For this to work, the random match model must have a negative score.

2) The time complexity of Smith-Waterman is $\Theta(n^2 m^2)$.

Example: Align HEAGAWGHEE and PAWHEAE.

Use BLOSUM 50 for substitution matrix and d=-8 for gap penalty.

(page 23 in the text)

| | | H | E | A | G | A | W | G | H | E | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | 0 | 0 | 0 | **0** | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 5 | 0 | **5** | 0 | 0 | 0 | 0 | 0 |
| W | 0 | 0 | 0 | 0 | 2 | 0 | **20** | **12** | 0 | 0 | 0 |
| H | 0 | 10 | 2 | 0 | 0 | 0 | 12 | 18 | **22** | 14 | 6 |
| E | 0 | 2 | 16 | 8 | 0 | 0 | 4 | 10 | 18 | **28** | 20 |
| A | 0 | 0 | 8 | 21 | 13 | 5 | 0 | 4 | 10 | 20 | 27 |
| E | 0 | 0 | 0 | 13 | 18 | 12 | 4 | 0 | 4 | 16 | 26 |



Heuristic alignment algorithms

- motivation: speed

  sequence DB ~ O(100,000,000) basepair

  query sequence 1000 basepair

  O(nm) time complexity => $10^{11}$ matrix cells in dynamic programming table

  if 10,000,000 cells/second => 10000 seconds ~ 3 hours.

- heuristic versus rigorous