

IN4030

Introduction to bioinformatics

Week 2

Sequence alignment

Torbjørn Rognes
torognes@ifi.uio.no

Department of Informatics, UiO
20 January 2021



UiO : Universitetet i Oslo

Overview of the lecture

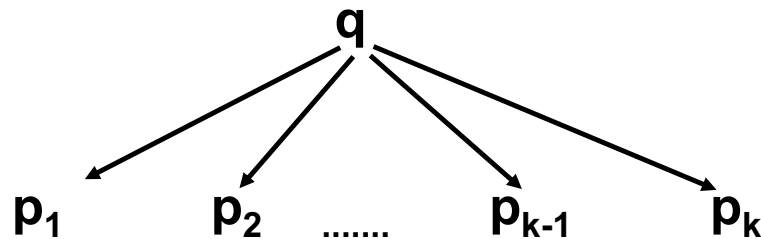
Parts of chapter 5 in the book

- Sequence alignment basics
- Edit distance
- Alignments and evolution
- Similarity vs homology
- Alignments and scoring systems
- Scoring matrices
- Gap penalty functions
- Global vs local alignment
- Pairwise and multiple alignment
- Graphical representation of alignments
- Optimal sequence alignments

Sequence alignment basics

Starting point:

- Sequences p_1, p_2, \dots, p_k (DNA, RNA, or protein)
- We have a hypothesis that the sequences (or parts of them) have developed during generations from a common unknown sequence q through a series of mutations, insertions and deletions in DNA



Goal:

- Identify the most likely residue by residue correspondance between the sequences. Residues may be amino acids or nucleotides.
- Determine whether the similarity between the sequences is significantly better than expected by chance

Edit distance

- The edit distance indicates the difference or dissimilarity between two strings
- The edit distance between two strings or sequences A and B is the total number of operations needed to transform the first string A into the second string B using the following operations:
 - Substitute one symbol with another
 - Delete a symbol
 - Insert a symbol
- The edit distance is also called the Levenshtein distance
- A low edit distance indicates that the two strings are similar
- Example (kitten -> sitting): 3 operations
 - kitten -> sitten (substitute k with s)
 - sitten -> sittin (substitute e with i)
 - sittin -> sitting (insert g at end)

Alignments and evolution

- We would like to identify the evolutionary relation between two sequences. What has happened during evolution?
- The sequences may be nucleotide sequences (DNA, RNA) or amino acid sequences (protein).
- We consider four different possible fates for a symbol during evolution:
 - No change of the symbol
 - Substitution of symbol a to symbol b ($a \rightarrow b$)
 - Deletion of symbol ($a \rightarrow$)
 - Insertion of symbol ($\rightarrow a$)
- Indel = insertion or deletion (when the direction of evolution is unknown)

Alignments and evolution: Example 1

Example with known history:

$q = \text{GLISVT}$, $d = \text{GIVT}$, $h = \text{GLVST}$

History of events:

GLISVT

GLIS-T (deletion of V)

GLVS-T (substitution of I to V)

GLV--T (deletion of S)

GIV--T (substitution of L to I)

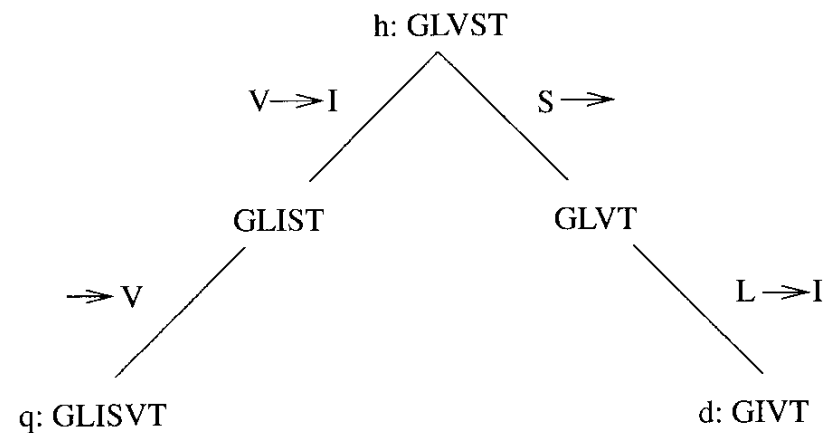


Figure 1.1 An evolution from h to q and d .

Alignments and evolution: Example 2

Example with unknown history

q= GLISVT, d=GIVT

Operations to consider (examples):

$I \leftrightarrow V$; $L \leftrightarrow I$; $V \leftrightarrow -$; $S \leftrightarrow -$

Possible history of events:

GLISVT

GLI-VT (deletion of S)

G-I-VT (deletion of L)

- The alignment may be different depending on what is known about the evolution of the sequences.
- In lack of additional information we choose the simplest explanation of the evolution (Occam's razor).

Similarity and homology

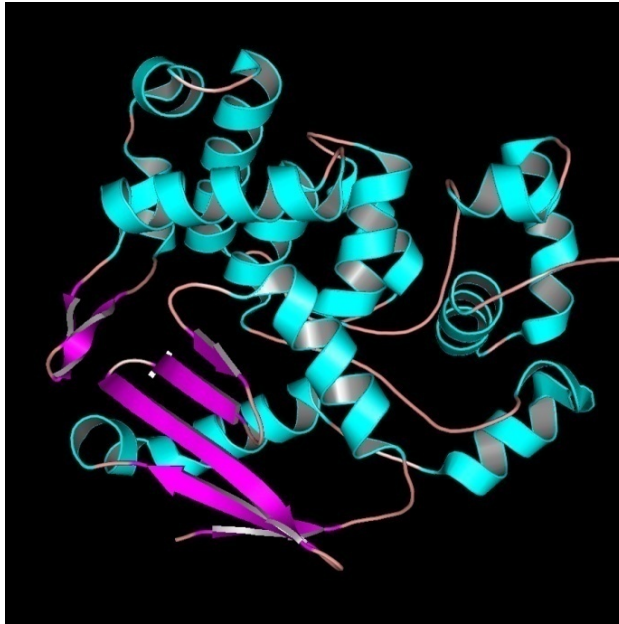
Two very important basic concepts:

- **Similarity:** Degree of likeness between two sequences, usually expressed as a percentage of similar (or identical) residues over a given length of the alignment. Can usually be easily calculated.
- **Homology:** Common evolutionary ancestry of two sequences. Can only be true or false. We can rarely be certain about this, it is therefore usually a hypothesis.

A high degree of similarity implies a high probability of homology

- If two sequences are very similar, the sequences are usually homologous
- If two sequences are not similar, we don't know if they are homologous
- If two sequences are not homologous, their sequences are usually not similar (but may be by chance)
- If two sequences are homologous, their sequences may or may not be similar; we don't know

Sequence similarity and protein evolution



E.coli AlkA

Hollis *et al.* (2000) *EMBO J.* **19**, 758-766 (PDB ID 1DIZ)



Human OGG1

Source: Bruner *et al.* (2000) *Nature* **403**, 859-866 (PDB ID 1EBM)

E.c.	AlkA	127	SVAMAAKL	TARVAQ	LYGERL	DDFPE--	YICFPT	PQRLAA	ADPQA-	LKALGM	PLKRAE	ALI	183
			++	+	+	+		+		+			+
H.s.	OGG1	151	NIARITG	MVERLC	QAFGPR	LIQLDD	VTYHGF	PSLQAL	AGPEVE	AHLRKL	GLGY-	RARYVS	209
E.c.	AlkA	184	HLANAAL	E-----	GTLPM	TIPGDV	EQAMKT	LQTFPG	IGRWTAN	YFAL			225
									+			+	+
H.s.	OGG1	210	ASARAILE	EQGGLA	WLQQLR	ESSYEE	AHKALC	ILPGVG	TKVADC	CICL			256

Sequence similarity and homology

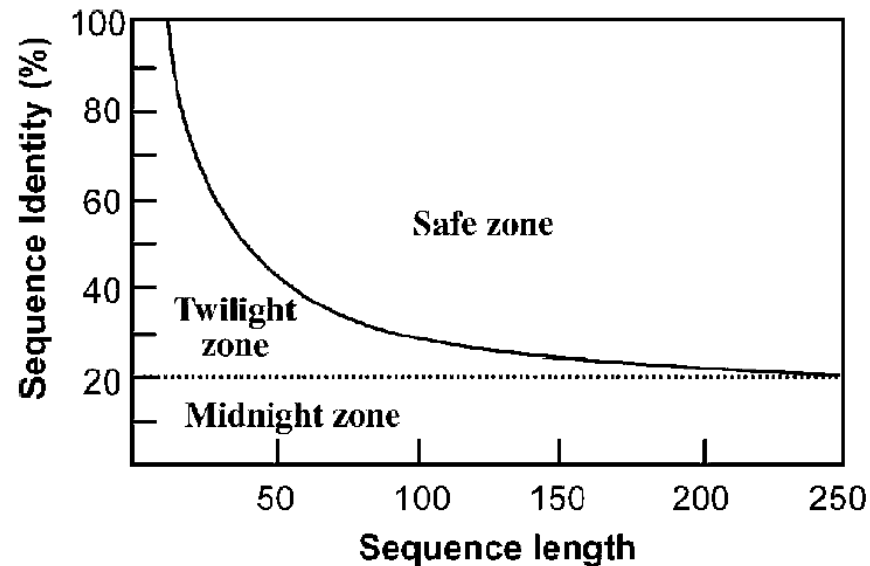


Figure 3.1: The three zones of protein sequence alignments. Two protein sequences can be regarded as homologous if the percentage sequence identity falls in the safe zone. Sequence identity values below the zone boundary, but above 20%, are considered to be in the twilight zone, where homologous relationships are less certain. The region below 20% is the midnight zone, where homologous relationships cannot be reliably determined. (Source: Modified from Rost [1999](#)).

Scoring system

- An alignment can be given a score
- The score indicates how well the two sequences are aligned
- The score of the best possible alignment of two sequences may indicate how similar they are
- The highest-scoring alignment is often the most correct alignment from an evolutionary view (or pretty close)
- For each column the following score is given:
 - A score for aligning symbols a and b with each other: R_{ab}
 - A score for aligning a blank/gap with a symbol: $-g$
This score is usually independent on the symbol and is always negative. It is called the gap penalty.
- The score for the entire alignment is equal to the sum of the scores from all columns in the alignment

A simple scoring system

Example: align sequences VEITGEIST and PRETERIST

$R_{ab} = 1$ if $a=b$ otherwise 0

$g = 1$

ALIGN1:

q'	:	V	-	E	I	T	G	E	I	S	T	
d'	:	P	R	E	-	T	E	R	I	-	T	
		0	-1	1	-1	1	0	0	1	-1	1	Score 1

ALIGN2:

q'	:	V	E	I	T	G	E	I	S	T	
d'	:	P	R	E	T	-	E	R	I	T	
		0	0	0	1	-1	1	0	0	1	Score 2

ALIGN3:

q'	:	-	V	E	I	T	G	E	-	I	S	T
d'	:	P	R	E	-	T	-	E	R	I	-	T
		-1	0	1	-1	1	-1	1	-1	1	-1	1
												Score 0

Scoring matrices

- Simple scoring systems with scores for matches and mismatches are usually too simplistic for real biological problems
- The values in a scoring matrix indicate the probability of a pair of symbols (nucleotides or amino acids) being evolutionary related and aligned in sequences of common evolutionary origin
- Many different matrices for DNA, RNA and proteins
- Usually symmetric: $R_{a,b} = R_{b,a}$
- Common matrices:
 - PAMxxx, e.g. PAM250 (PAM=Procent/Point Accepted Mutation)
 - BLOSUMxx, e.g. BLOSUM62 (BLOcks of Amino Acid SUBstitution Matrix)

Common alignment scoring system

- Substitution score matrix
 - Score for aligning any two residues to each other
 - Identical residues have large positive scores
 - Similar residues have small positive scores
 - Very different residues have large negative scores
- Gap penalties
 - Penalty for opening a gap in a sequence (Q)
 - Penalty for extending a gap (R)
 - Typical gap function: $G = Q + R * L$, where L is length of gap
 - Example: Q=11, R=1

BLOSUM62 amino acid substitution score matrix																				
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-2	-1	1	0	-3	-2	0	
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	0	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	-2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

```

E.c. AlkA 127 SVAMAAKLTARVAQLYGERLDDFPE--YICFPTPQRLAAADPQA-LKALGMPLKRAEALI 183
              ++|      +  |+ | +| ||      +  |  ||+ | ||  + +| |+ ||+  ||  +
H.s. OGG1 151 NIARITGMVERLCQAFGPRLIQLDDVTYHGFPSLQALAGPEVEAHLRKLGLGY-RARYVS 209

E.c. AlkA 184 HLANAALE-----GTLPMTIPGDVEQAMKTLQTFPGIGRWTANYFAL 225
              | | ||      |      |+| | |  ||+|  |+  |
H.s. OGG1 210 ASARAILEEQGGLAWLQQQLRESSYEEAHKALCILPGVGTKVADCICL 256
    
```

The BLOSUM62 substitution score matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Example scoring matrix: PAM250

Table 1.1 Scoring matrix for the evolutionary distance of 250 PAM, rounded to one digit.
The amino acids occur in alphabetic order of their full names.

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

Score matrix for DNA

- For DNA, since we only have 4 nucleotides, the score matrix is simple.
 - Equal penalties for all substitutions (mismatches) (Jukes and Cantor model) (most common)
 - Different (usually higher) penalties for transversions (A,G \leftrightarrow C,T) than transitions (A \leftrightarrow G and C \leftrightarrow T) (Kimura model)

	A	C	G	T
A	5	-4	-4	-4
C	-4	5	-4	-4
G	-4	-4	5	-4
T	-4	-4	-4	5

Jukes and Cantor

	A	C	G	T
A	1	-5	-1	-5
C	-5	1	-5	-1
G	-1	-5	1	-5
T	-5	-1	-5	1

Kimura

Gap / Indels

- Gaps may have different lengths
- One gap may consist of several "indels", which are single insertions or deletions

- Example:

AC--GRTV

ACMTG-TV

- Longer gaps should have a higher penalty than shorter gaps

Gap penalty functions

- It is difficult to score or penalise gaps correctly
- The biological reasoning for gaps is difficult
- Gaps of length l gives a penalty g_l
- Examples of different gap penalty functions:
 - Constant gap penalty: $g_l = k$
 - Linear gap penalty: $g_l = g * l$
 - Affine gap penalty: $g_l = g_{\text{open}} + l * g_{\text{extend}}$
Note that in some contexts the following is used: $g_l = g_{\text{open}} + (l-1)*g_{\text{extend}}$
 - Logarithmic gap penalty: $g_l = g_{\text{open}} + \log l$
- Concave gap penalty:
 - $g_l \leq g_{l-r} + g_r$ for all r where $0 < r < l$
 - Concave gap penalties are biologically meaningful because it gives less (or equal) penalty for one long gap than for two or more gaps of the same total length
- Sometimes it may be natural for gaps at the ends of the sequences to be penalised less (or not at all) (semi-global alignment)

Graphs of gap penalty functions

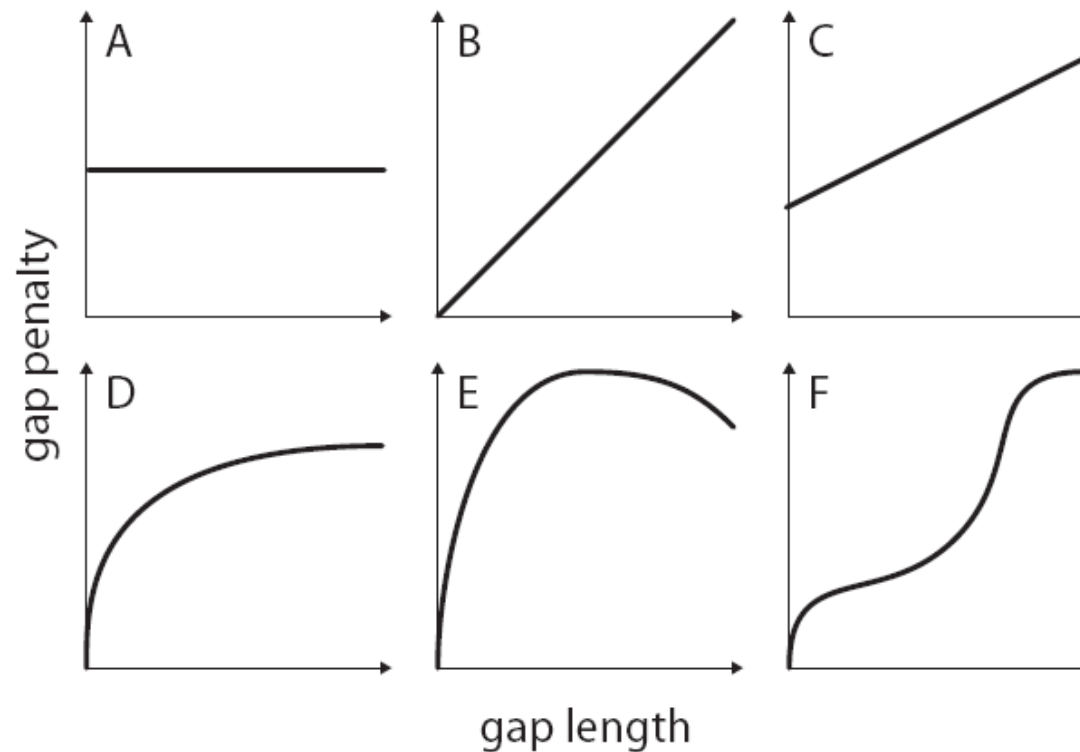


Figure 8: Examples of gap penalty functions

Graphs of constant (A), proportional (B), affine (C), logarithmic (D), concave (E) and monotonic (F) gap penalty functions are shown.

Example - different gap penalty functions

Example

A possible alignment of the insulin proteins from sheep and zebrafish is

```
Fish: MAVWLQAGALLVLLVV-SSVSTNPGTPQHLCGSHLVDALYLVCGPTGFFYNPK--R
Sheep: MALWTRLVPLLALLALWAPAPAHAFVNQHLCGSHLVEALYLVCGERGFFYTPKARR

Fish: DVE-PLLGFLPPKSAQETEVADFAFKDHAELIRKRGIVEQCCHKPCSIFELQNYCN
Sheep: EVEGPQVGAL--ELAGGPG-AG-GL-EGPP-Q-KRGIVEQCCAGVCSLYQLENYCN
```

The alignment of the insulin proteins in Section 1.2 was found by using the PAM 250 matrix, and a linear gap penalty of $g = 5$. It has nine gaps. If we change to an affine gap penalty, $5 + (l - 1)0.5$ we get the alignment:

```
MAVWLQAGALLVLLVV-SSVSTNPGTPQHLCGSHLVDALYLVCGPTGFFYNPK--RDVE-PLL
MALWTRLVPLLALLALWAPAPAHAFVNQHLCGSHLVEALYLVCGERGFFYTPKARREVEGPQV

GFLPPKSAQETEVADFAFKDHAELIRKRGIVEQCCHKPCSIFELQNYCN
GALELAGGPGAG----GLEGPPQ---KRGIVEQCCAGVCSLYQLENYCN
```

which contains fewer (five) gaps.

Changing the gap penalty to $1 + (l - 1)0.1$ results in the alignment:

```
MALWTRL-V-PLLALL---ALWA--P-APAHAFVNQHLCGSHLVEALYLVCGPTGFFYNPK--R
MAVW--LQAGALLVLLVVSSV-STNPGTP-----QHLCGSHLVDALYLVCGERGFFYTPKARR

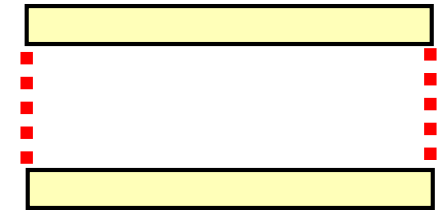
DVE-P----L-LAGGPGAGG-LEGPP---Q-----FAFKDHAELIRKRGIVEQCCHKP--CSI
EVEGPQVGAL-EL-----GFL--PPKSAQETEVAD-----KRGIVEQCC--AGVCSL

FELQNYCN
YQLENYCN
```

Global and local alignments

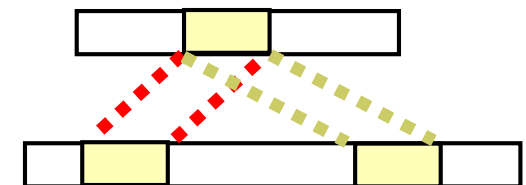
Global alignment:

- Alignment of entire sequences (all symbols)
- May be used when the sequences are of approximately equal length and are expected to be related over their entire length.



Local alignment:

- Alignment of subsequences from each sequence
- Part of the problem is to identify which parts of the sequences should be included
- Is used when the sequences are of unequal length; and/or only certain regions in the sequences are assumed to be related (conserved domains).



Global and local alignments

Figure 3.2: An example of pairwise sequence comparison showing the distinction between global and local alignment. The global alignment (*top*) includes all residues of both sequences. The region with the highest similarity is highlighted in a box. The local alignment only includes portions of the two sequences that have the highest regional similarity. In the line between the two sequences, “:” indicates identical residue matches and “.” indicates similar residue matches.

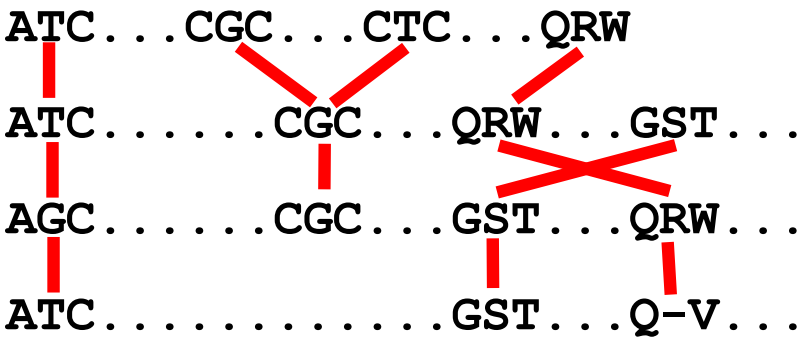
```
seq1  EARDF-NQYYSSIKRSGSIQ
      . : .:.:.:.:.:.
seq2  LPKLFIDQYYSSIKRTMG-H
```

global sequence alignment

```
seq1  NQYYSSIKRS
      .:.:.:.:.
seq2  DQYYSSIKRT
```

local sequence alignment

Types of alignments

	Global	Local
Pairwise	GPSS--QTGKGS-SR GQSTIKQSGKRSWSMGSTWQRV.....GS-WQTV.....
Multiple	ATCTCGA--GA ATC-CGA--GA ATGTCGAC-GA ATGTCGACAGA AT-TCAAC-GA	 <p> ATC...CGC...CTC...QRW ATC.....CGC...QRW...GST... AGC.....CGC...GST...QRW... ATC.....GST...Q-V... </p>

Definition of a global sequence alignment

- An alignment is a arrangement of two sequences ***q*** and ***d*** that indicates which pairs of symbols, one from each sequence, that belongs together and corresponds to each other
- All symbols in ***q*** and ***d*** must be included in the alignment and in the same order as in the original sequences
- Symbols in ***q*** and ***d*** cannot be aligned with more than one symbol from the other sequence
- A symbol from ***q*** or ***d*** may also be aligned with a blank symbol indicated by a hyphen ("-") (indel) (gap)
- To blank symbols (gaps) cannot be aligned.

Example

A possible alignment of the insulin proteins from sheep and zebrafish is

```
Fish:  MAVWLQAGALLVLLVV-SSVSTNPGTPQHLCGSHLVDALYLVCGPTGFFYNPK--R
Sheep: MALWTRLVPLLALLALWAPAPAHAFVNQHLCSHLVEALYLVCGERGFFYTPKARR

Fish:  DVE-PLLGFLPPKSAQETEVADFAFKDHAELIRKRGIVEQCCHKPCSIFELQNYCN
Sheep: EVEGPQVGAL--ELAGGPG-AG-GL-EGPP-Q-KRGIVEQCCAGVCSLYQLENYCN
```

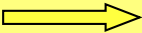
Example, global alignments

Assume that we have the following sequences:

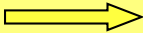
x = AWDIY

y = SWJKLY

A global alignment of **x** and **y** of length 7:

A W - D - I Y		x* = AW-D-IY
S W J K L - Y		y* = SWJKL-Y

A global alignment of length 9:

- A - W - D - I Y		x* = -A-W-D-IY
S - W - J K L - Y		y* = S-W-JKL-Y

Graphical representation of alignments

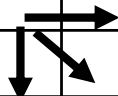
Assume that we are going to align these sequences:

IPLWRQK

VPWRSA

We could create the following table:

	?	I	P	L	W	R	Q	K
?								
V								
P								
W								
R								
S								
A								



Definition:

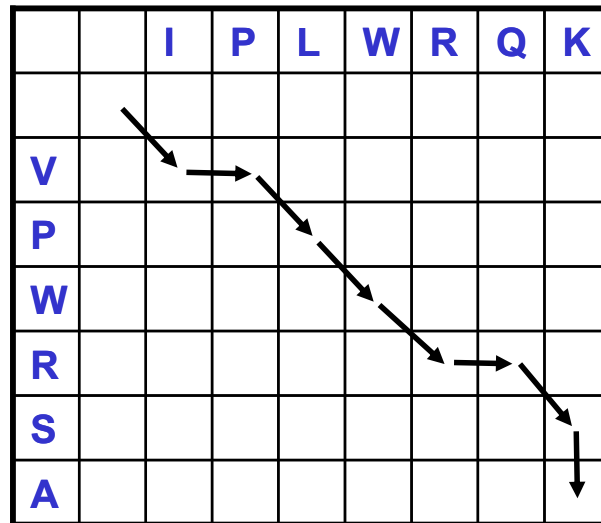
By residue is meant a symbol which is not the gap symbol.

An alignment of the sequences can be described as a series of moves – one way – from the upper left corner to the bottom right corner. Allowed moves:

Move	Alignment
right	residue – gap
down	gap – residue
diagonally	residue – residue

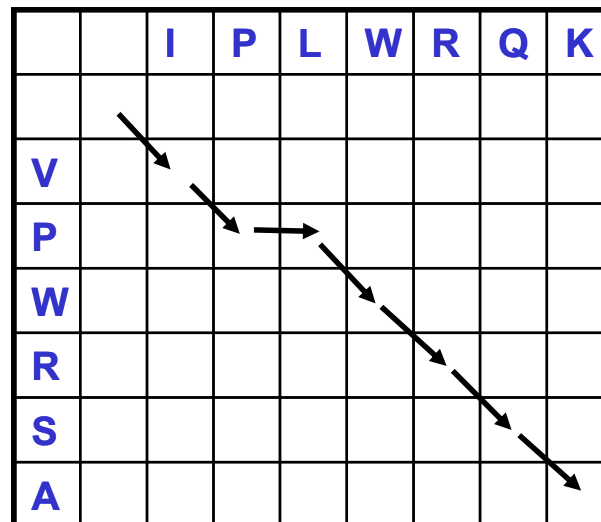
Examples

		I	P	L	W	R	Q	K
V								
P								
W								
R								
S								
A								



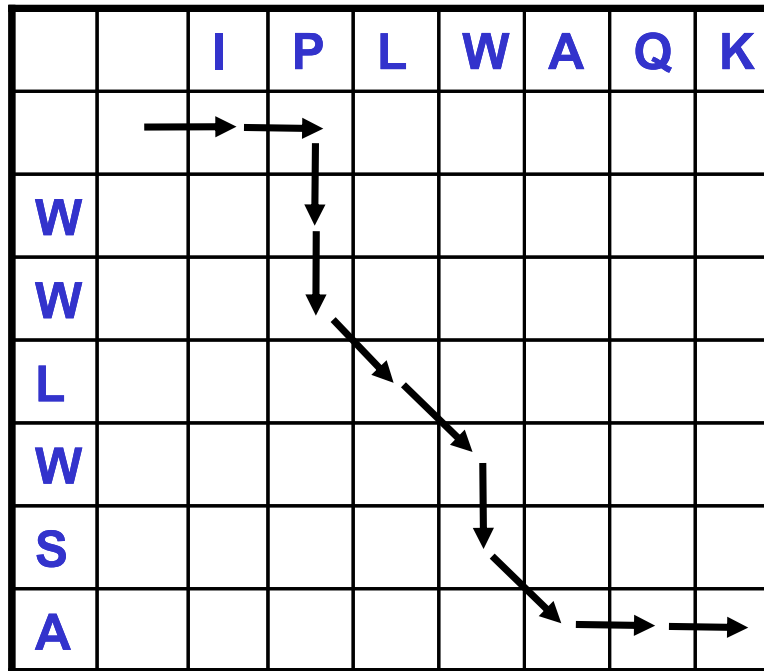
I P L W R Q K -
V - P W R - S A

		I	P	L	W	R	Q	K
V								
P								
W								
R								
S								
A								



I P L W R Q K
V P - W R S A

Example



Show the alignment that corresponds to the graph above.

```

I  P  -  -  L  W  -  A  Q  K
-  -  W  W  L  W  S  A  -  -
  
```

Example

We have the following sequence alignment:

P - - C G N - K V D
P D C - - N E K V -

Fill in the table below corresponding to the alignment.

	?	P	C	G	N	K	V	D
?								
P								
D								
C								
N								
E								
K								
V								

Alignments = paths

It is intuitively clear from what we have seen that:

- To each pairwise alignment there exists one and only one path through the table
- For each path through the table, there exists one and only one pairwise alignment

Thus:

Finding an optimal alignment of the sequences is equivalent to finding an optimal path through the table.

Optimal alignments

- Let $S(\dots)$ be a chosen score function for an alignment.
- Let \mathbf{x} and \mathbf{y} be two (not necessarily equally long) sequences
- An optimal global alignment of (\mathbf{x}, \mathbf{y}) is a global alignment having the maximum possible score $S(\mathbf{x}^*, \mathbf{y}^*)$ of any global alignment of \mathbf{x} and \mathbf{y} .
- There are often several different optimal alignments (all having the same score).

Brute force algorithm

Brute force algorithm to identify all optimal global alignments:

- Identify all possible global alignments of (x,y)
- Compute the score for each of them
- Find the alignment(s) with the highest score

To carry out this in reality we need:

- 1) a method to generate all possible global alignments
- 2) a method to compute all the alignment scores in reasonable time

As we will see, 1 is easy, but 2 is very hard.

Brute force is impractical

- The Brute force algorithm is unusable for anything but very short sequences:

Length of x and y	Number of alignments \geq
10	8097453
50	$1.5 * 10^{37}$
100	$2.1 * 10^{75}$

- More efficient: use dynamic programming.

Alignment algorithms

	Global	Local
Pairwise	Dynamic programming (Needleman-Wunsch)	Dynamic programming (Smith-Waterman) FASTA, BLAST, ...
Multiple	Dynamic programming (Carillo-Lipman) ClustalW, T-Coffee, MAFFT, Muscle, ...	Dynamic programming (Carillo-Lipman) MEME, ...

Note: dynamic programming gives an optimal solution, but is usually extremely space and time demanding with many sequences ($n > 2$)

How to find the best alignment(s)?

- There are too many possible alignments of two sequences to enable examination of every possible alignment
- There is an algorithm to identify the best (optimal) alignment(s), i.e the one(s) with the highest score
- The algorithm is of the dynamic programming (DP) type
- The algorithm was initially described by Needleman and Wunsch in 1970
- Two steps:
 - First, identify the highest possible score using DP
 - Then, identify the alignment(s) with the highest score (using temporary results from the initial step)
- Dynamic programming:
 - General method for solving recursive problems by storing temporary results from smaller problems along the way
 - Used to solve many problems in bioinformatics