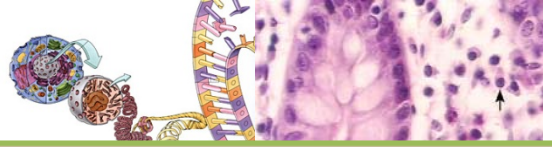



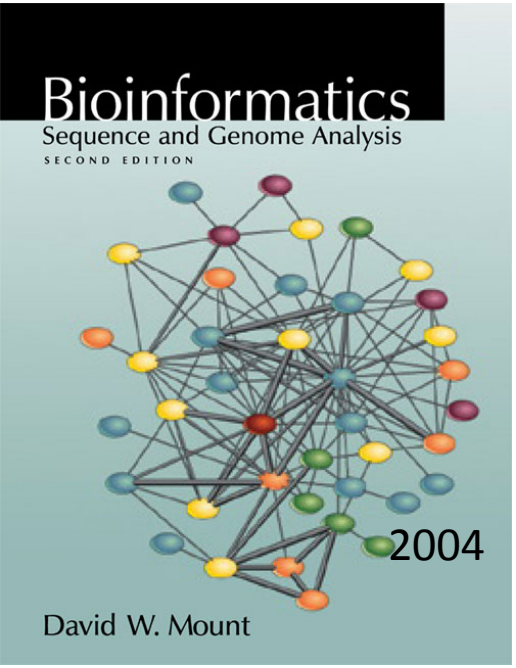


Kusuma School of Biological Sciences

SBL-100: Introductory Biology for Engineers

L-4

Sequence Alignment

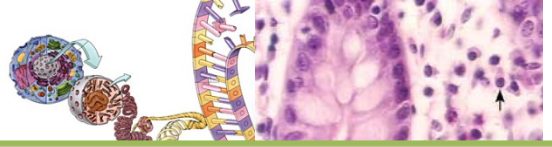


Similarity searches

Girl-1: This rose is amazingly beautiful.
Boy-1: Yes, I have seen many but this one is out of the ordinary.

Boy-2: Her prose is truly amazing.
Girl-2: Yes ,I have read many but this one is extraordinary.

Computer, based on similarity searches, pairs Girl-1 with Boy-2 and Boy-1 with Girl-2. Do you agree?



Kusuma School of Biological Sciences

The Nobel Prize in Chemistry 1958 was awarded to Frederick Sanger *"for his work on the structure of proteins, especially that of insulin"*.

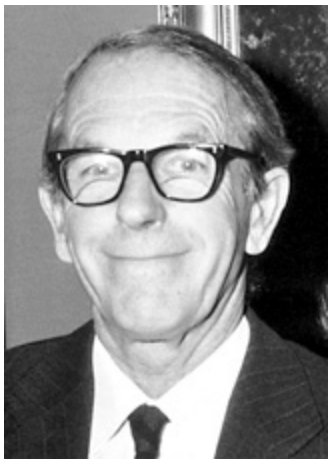
By structure above is meant, covalent connectivity viz. sequence in today's parlance.

Protein Sequencing

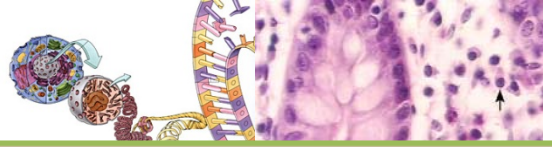
Frederick Sanger, 1958

The Nobel Prize in Chemistry 1980 was divided, one half awarded to Paul Berg *"for his fundamental studies of the biochemistry of nucleic acids, with particular regard to recombinant-DNA"*, the other half jointly to Walter Gilbert and Frederick Sanger *"for their contributions concerning the determination of base sequences in nucleic acids"*.

DNA Sequencing



Frederick Sanger, 1980



Kusuma School of Biological Sciences

Sequence Alignment

Why Align?

Useful for discovering structural, functional & evolutionary relationships in biological sequences.

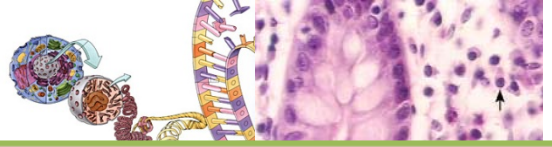
To identify functionally important sites
To demonstrate homology between sequences
To establish molecular phylogeny
To detect weak but significant similarities in sequence databases
To facilitate structure prediction (eg. Secondary and tertiary structure of proteins)
To facilitate function prediction
To design primers for PCR

Three Steps to alignment

1.Search for homologues in sequence databases
2.Compute alignments
3.Check and edit alignments

Basic Concept

Homology. Two sequences are said to be homologous if they derive from a common ancestor. Homology is inferred from sequence similarity, Sequence similarity does not guarantee homology.



Alignment Types: Sequence alignment can be global or local

Global alignment: Align the entire length of the sequences to match as many letters (amino acids or nucleic acid bases as appropriate) as possible. Sequences that are quite similar & of same length are suitable candidates.

Local alignment: A stretch of sequence is matched, thus more suitable for aligning sequences that share a conserved region or domain or for sequences that differ in length.

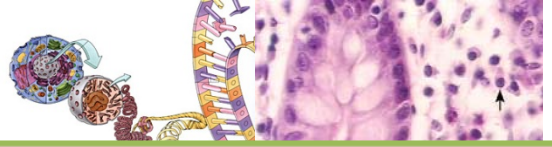
```
G P S S Q T W G K G S S R --- -I W D N S Y T
|                 | | |                 |       |
G N Q T I R  T  G K G A W Q I M R D G D Y A
```

(Global Alignment)

```
------------------------------S S R K G ----------------
                               | | | | |
 ------------------------------S S R K G ----------------
```

(Local Alignment)

Vertical bars match identical amino acids (or bases) or a stretch of amino acids (or bases).

Methods for alignment:

Two kinds of alignment – comparing two sequences (Pairwise) or more (Multiple sequence alignment) either by character matching or pattern matching.

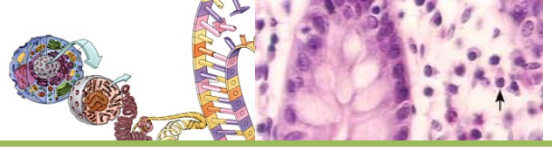
In alignments, three types of mutations are considered: Substitutions, insertions & deletions (indels)

Pair wise Alignments:

Dot Matrix Analysis
Dynamic Programming (DP)
Fast word or k-tuple methods like FASTA, BLAST

Multiple Sequence alignment:

Optimal global alignments - MSA
Progressive Methods of global multiple sequence alignment – CLUSTALW,PILEUP,MAP,MULTALIN
Iterative global alignment methods – Genetic algorithm(SAGA), HMM methods(HMMR)



Needleman & Wunsch algorithm for global alignment

(Dynamic programming method)

Ref.: Attwood & Parry-Smith: Introduction to Bioinformatics

Step 1. Consider aligning the following two sequences

1. ALCDRYFQ & 2. NCDRYYQ

Step 2: Construct a two dimensional matrix and enter a zero for a mismatch and 1 for a match in each cell.

	1. A	2. L	3. C	4. D	5. R	6. Y	7. F	8. Q
1. N	0	0	0	0	0	0	0	0
2. C	0	0	1	0	0	0	0	0
3. D	0	0	0	1	0	0	0	0
4. R	0	0	0	0	1	0	0	0
5. Y	0	0	0	0	0	1	0	0
6. Y	0	0	0	0	0	1	0	0
7. Q	0	0	0	0	0	0	0	1

Step3. Start with the bottom rightmost cell (last cell of the last column)(in this case this happens to be a Q-Q pair in the 7^{th} row and 8^{th} column with a value of 1) and proceed upwards to add the values in each cell and reconstruct the matrix in the following manner. In this the values in the last row (i.e. 7th row) and last column (i.e. 8^{th} column) are untouched.

Step 4. From the last cell, move one column to the left (i.e. to the 7th column) and add the current value of each cell in each row (leaving the last row) as shown below.

	A	L	C	D	R	Y	F	Q
N							1	0
C							1	0
D							1	0
R							1	0
Y							1	0
Y							1	0
Q							0 ←	**1**

Step 5. Now move one more column to the left (i.e. to the 6th column) and repeat the process.

	1. A	2. L	3. C	4. D	5. R	6. Y	7. F	8. Q
1. N						1	1	0
2. C						1	1	0
3. D						1	1	0
4. R						1	1	0
5. Y						2	**1**	**0**
6. Y						2	**1**	**0**
7. Q						0 ←	**0** ←	**1**

While entering the sums in the 5th column, the value of the previous sum is added to the value in each cell. The value of the previous sum is judged by looking at the block diagonal matrix on the lower right half. For instance consider the 4th row and 6th column i.e. the (R,Y) pair whose individual cell value is zero (0). To enter the sum in this (4,6) matrix element, find out the maximum value occurring in the 5th to last row and 7th to last column which in this case happens to be one (1). This when added to the current value of the cell (0) gives (1). Similarly, the sum value in the matrix element (5,6) is 1+1 (2).

Step 6. Some intermediate entries up to the 4th column are as follows.

	1. A	2. L	3. C	4. D	5. R	6. Y	7. F	8. Q
1. N				3	2	1	1	0
2. C				3	2	1	1	0
3. D				4	2	1	1	0
4. R				2	**3**	**1**	**1**	**0**
5. Y				2	**2**	**2**	**1**	**0**
6. Y				1	**1**	**2**	**1**	**0**
7. Q				0←	**0←**	**0 ←**	**0 ←**	**1**

Illustrating the procedure again, the value for the matrix element [3,4] is 1 for a D,D match. The maximum value of the sum in the lower block diagonal matrix (in the lower right) is 3. The sum value to be entered in [3,4] thus is 3+1 (4).

Step 7. Proceeding as indicated above we arrive at the following sum matrix.

	1. A	2. L	3. C	4. D	5. R	6. Y	7. F	8. Q
1. N	5	5	4	3	2	1	1	0
2. C	4	4	5	3	2	1	1	0
3. D	3	3	3	4	2	1	1	0
4. R	2	2	2	2	3	1	1	0
5. Y	2	2	2	2	2	2	**1**	**0**
6. Y	1	1	1	1	1	2	**1**	**0**
7. Q	0	0	0	0	0	0 ←	**0** ←	**1**

Step 8. There are 5 matches. The highest value of 5 in the sum matrix is obtained at the element [2,3] indicating that a possible alignment of 3rd residue in sequence 1 with 2nd residue in sequence 2. (this has a score of 5). Considering the next residue aligned i.e. [3,4] which has a score of 4 is also the best alignment up to that point from right.

```
A L C D R Y F Q
    | | | |   |
N C D R Y Y Q
```

Local alignment

Smith-Waterman algorithm (Dynamic programming method)

Ref.: Attwood & Parry-Smith: Introduction to Bioinformatics

(Regions of local similarity – may be no overall satisfactory alignment - a sensitive technique)

Step 1. Consider aligning the following two sequences

1. ALCDRYFQ & 2. NCDRYYQ

Step 2: Construct a two dimensional matrix and enter a zero (floating points) for the all the edge elements (i.e. all elements in the 1st row and 1st column)

	x	1. A	2. L	3. C	4. D	5. R	6. Y	7. F	8. Q
x	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1. N	0.0								
2. C	0.0								
3. D	0.0								
4. R	0.0								
5. Y	0.0								
6. Y	0.0								
7. Q	0.0								

Step 3: To the value of the upper left diagonal element (diagonal predecessor i-1, j-1 element), and proceeding diagonally, add 1 for match; -0.33 for a mismatch and enter the sum in the cell under consideration. Negative numbers are replaced by zero. The numbers are rounded off to the first decimal.

	x	1. A	2. L	3. C	4. D	5. R	6. Y	7. F	8. Q
x	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1. N	0.0	0.0	0.0						
2. C	0.0		0.0	1.0					
3. D	0.0			0.0	2.0				
4. R	0.0				0.0	3.0			
5. Y	0.0					0.0	4.0		
6. Y	0.0						1.0	3.7	
7. Q	0.0							0.7	4.7

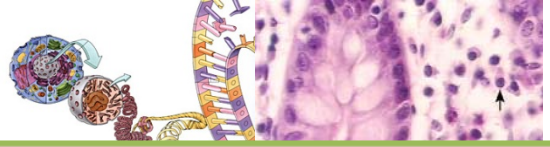
Step 4. Continuing further, the following matrix is constructed.

	x	1. A	2. L	3. C	4. D	5. R	6. Y	7. F	8. Q
x	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1. N	0.0	0.0	**0.0**	0.0	0.0	0.0	0.0	0.0	0.0
2. C	0.0	0.0	0.0	**1.0**	0.0	0.0	0.0	0.0	0.0
3. D	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	0.0
4. R	0.0	0.0	0.0	0.0	0.0	**3.0**	0.0	0.0	0.0
5. Y	0.0	0.0	0.0	0.0	0.0	0.0	**4.0**	0.0	0.0
6. Y	0.0	0.0	0.0	0.0	0.0	0.0	1.0	**3.7**	0.0
7. Q	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	**4.7**

The highest score is located representing the end point of the highest scoring alignment. Other elements leading to this cell are determined by a backtracking procedure.

```
A L C D R Y F Q
    |   |   |     |
N C D R Y Y Q
```

Lower right bottom element (N-terminus) does not have to carry the maximum value.



Heuristic/k-tuple methods:

FASTA (Pearson & Lipman, 1988) provides a rapid way to find short stretches (words) of similar sequence between a query sequence and any reference sequence in the database.

BASIC LOCAL ALIGNMENT TOOL (BLAST) (One of the most widely used softwares for sequence alignments)

BLAST (Altschul, 1990) Identifies common words in two sequences and identifies which of these words are more significant such that they are good indicators of similarity between two sequences…

Multiple Sequence Alignment (MSA)…(to build the tree of life, *inter alia*…by comparing thousands of genome/protein sequences - each containing hundreds of amino acids or thousands/millions/billions of base pairs - by aligning each with every other sequence…)

Assignment 8 (2020)

"The computation of an exact MSA is NP-Complete and therefore impossible for all but unrealistically small datasets (Wang & Jiang, 1994). MSA computation therefore depends on approximate algorithms or heuristics and it is worth mentioning that almost every conceivable optimization technique has been adapted into a heuristic multiple sequence aligner…" (Kemena & Notredame, 2009). Can you devise a reliable algorithm to do MSA in seconds/minutes?