International Journal of PharmTech Research
CODEN (USA): IJPRIF, ISSN: 0974-4304
Vol.8, No.5, pp 813-817, 2015

# Theoretical Determination of Amino Acid Substitution Groups Using Binary String

**M. Yamuna**

**SAS, VIT University, Vellore, Tamilnadu, India – 632 014**

**Abstract:** Proteins function at the right time and at the right place for various cells to function properly. Changes in gene instructions causes severe medical problems. These changes can be identified by protein alignment and its substitution groups. Protein alignments reflects various properties, which aid in identifying the cause of the problems. This paper introduces a method for theoretical identification of amino acid substitution groups. This method approaches to view amino acid substitution as a pair wise phenomenon and characterizes it using binary matrix. Amino acids satisfy various properties. It cannot be decided which property is most important in classifying and determining protein structure. Based on the existing method of multi property based classification, the proposed binary matrix is created to identify amino acid pairs so that their pair wise property score is atleast 50 percent.
**Keywords:** Amino acid, Protein, Binary String, Substitution grou.

## Introduction

There are two popular trends in sequence analysis. One trend focuses primarily on applying rigorous mathematical methods to bring out the optimal alignment of the sequences, thus leading to revelation of possible hidden biological significance between sequences. The other trend stretches on correctly identifying the actual biological significance between the sequences, where some or all biological features may have already been known. These two trends emerge from specific tasks bioinformatics scientists are dealing with.

The first trend relates to predicting the sequence structures and homology, species evolution, or determine the relationship between sequences in order to categorizing and organizing sequence databases [1].

In this paper, two different approaches to sequence alignment have been discussed and compared. The first method employs Boolean algebra which is a two-valued logic whereas the second is based on Fuzzy logic which is a multi-valued logic [2].

In [3] several distance measures are compared and examine a method that involves circular shifting one sequence against the other for finding good alignment to minimize Hamming distance. Sandeep Hosangadi also uses run-length encoding together with LZ77 to characterize information in a binary sequence. Mathematics and computer science play an effective role in multiple sequence alignments. Various techniques are determined and used extensively for protein sequence alignment. In this paper we propose a method of protein sequence alignment using binary numbers.

### Protein Sequence Alignment

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary

relationships between the sequences.[1]Aligned sequences of nucleotides or amino acid residues are typically represented as rows within a matrix. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns [4].

In [5] Kristine Yu introduced a novel method for theoretical determination of amino acid substitution groups. The method here involves making a binary matrix based on 48 qualitative physicochemical properties and calculating a substitution matrix based on this using dot products. Isolated groups with high scores are determined to be valid substitution groups and conserved groups are derived from these valid groups. 258 valid groups and 31 conserved groups are found. Based on this discussion the normalized matrix of substitution scores is as seen in Table – 1.

**Normalized matrix of substitution scores**

**Table – 1**

| | A | R | N | D | C | E | Q | G | H | I | L | K | M | F | P | S | T | W | Y | V | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1.00 | 0.13 | 0.38 | 0.29 | 0.63 | 0.21 | 0.29 | 0.75 | 0.21 | 0.71 | 0.79 | 0.17 | 0.71 | 0.58 | 0.63 | 0.50 | 0.29 | 0.33 | 0.17 | 0.75 | A |
| R | 0.13 | 1.00 | 0.42 | 0.42 | 0.25 | 0.50 | 0.50 | 0.13 | 0.58 | 0.17 | 0.25 | 0.96 | 0.33 | 0.29 | 0.17 | 0.29 | 0.17 | 0.38 | 0.29 | 0.13 | R |
| N | 0.38 | 0.42 | 1.00 | 0.75 | 0.50 | 0.58 | 0.83 | 0.38 | 0.58 | 0.33 | 0.42 | 0.46 | 0.50 | 0.38 | 0.50 | 0.63 | 0.58 | 0.46 | 0.38 | 0.46 | N |
| D | 0.29 | 0.42 | 0.75 | 1.00 | 0.42 | 0.83 | 0.58 | 0.29 | 0.58 | 0.25 | 0.33 | 0.46 | 0.42 | 0.29 | 0.42 | 0.46 | 0.42 | 0.29 | 0.29 | 0.38 | D |
| C | 0.63 | 0.25 | 0.50 | 0.42 | 1.00 | 0.33 | 0.42 | 0.46 | 0.42 | 0.42 | 0.50 | 0.29 | 0.67 | 0.46 | 0.42 | 0.63 | 0.42 | 0.38 | 0.38 | 0.46 | C |
| E | 0.21 | 0.50 | 0.58 | 0.83 | 0.33 | 1.00 | 0.75 | 0.21 | 0.58 | 0.25 | 0.33 | 0.54 | 0.42 | 0.29 | 0.25 | 0.38 | 0.25 | 0.29 | 0.29 | 0.21 | E |
| Q | 0.29 | 0.50 | 0.83 | 0.58 | 0.42 | 0.75 | 1.00 | 0.29 | 0.58 | 0.33 | 0.42 | 0.54 | 0.50 | 0.38 | 0.33 | 0.54 | 0.42 | 0.46 | 0.38 | 0.29 | Q |
| G | 0.75 | 0.13 | 0.38 | 0.29 | 0.46 | 0.21 | 0.29 | 1.00 | 0.21 | 0.46 | 0.54 | 0.17 | 0.54 | 0.42 | 0.54 | 0.50 | 0.29 | 0.25 | 0.08 | 0.50 | G |
| H | 0.21 | 0.58 | 0.58 | 0.58 | 0.42 | 0.58 | 0.58 | 0.21 | 1.00 | 0.25 | 0.33 | 0.63 | 0.42 | 0.38 | 0.25 | 0.46 | 0.33 | 0.54 | 0.46 | 0.21 | H |
| I | 0.71 | 0.17 | 0.33 | 0.25 | 0.42 | 0.25 | 0.33 | 0.46 | 0.25 | 1.00 | 0.92 | 0.21 | 0.75 | 0.63 | 0.58 | 0.29 | 0.42 | 0.38 | 0.21 | 0.88 | I |
| L | 0.79 | 0.25 | 0.42 | 0.33 | 0.50 | 0.33 | 0.42 | 0.54 | 0.33 | 0.92 | 1.00 | 0.29 | 0.83 | 0.71 | 0.67 | 0.38 | 0.33 | 0.46 | 0.29 | 0.88 | L |
| K | 0.17 | 0.96 | 0.46 | 0.46 | 0.29 | 0.54 | 0.54 | 0.17 | 0.63 | 0.21 | 0.29 | 1.00 | 0.38 | 0.33 | 0.21 | 0.33 | 0.21 | 0.42 | 0.33 | 0.17 | K |
| M | 0.71 | 0.33 | 0.50 | 0.42 | 0.67 | 0.42 | 0.50 | 0.54 | 0.42 | 0.75 | 0.83 | 0.38 | 1.00 | 0.79 | 0.67 | 0.46 | 0.33 | 0.54 | 0.38 | 0.71 | M |
| F | 0.58 | 0.29 | 0.38 | 0.29 | 0.46 | 0.29 | 0.38 | 0.42 | 0.38 | 0.63 | 0.71 | 0.33 | 0.79 | 1.00 | 0.54 | 0.33 | 0.21 | 0.67 | 0.58 | 0.58 | F |
| P | 0.63 | 0.17 | 0.50 | 0.42 | 0.42 | 0.25 | 0.33 | 0.54 | 0.25 | 0.58 | 0.67 | 0.21 | 0.67 | 0.54 | 1.00 | 0.38 | 0.33 | 0.46 | 0.29 | 0.71 | P |
| S | 0.50 | 0.29 | 0.63 | 0.46 | 0.63 | 0.38 | 0.54 | 0.50 | 0.46 | 0.29 | 0.38 | 0.33 | 0.46 | 0.33 | 0.38 | 1.00 | 0.79 | 0.50 | 0.50 | 0.33 | S |
| T | 0.29 | 0.17 | 0.58 | 0.42 | 0.42 | 0.25 | 0.42 | 0.29 | 0.33 | 0.42 | 0.33 | 0.21 | 0.33 | 0.21 | 0.33 | 0.79 | 1.00 | 0.38 | 0.38 | 0.46 | T |
| W | 0.33 | 0.38 | 0.46 | 0.29 | 0.38 | 0.29 | 0.46 | 0.25 | 0.54 | 0.38 | 0.46 | 0.42 | 0.54 | 0.67 | 0.46 | 0.50 | 0.38 | 1.00 | 0.75 | 0.33 | W |
| Y | 0.17 | 0.29 | 0.38 | 0.29 | 0.38 | 0.29 | 0.38 | 0.08 | 0.46 | 0.21 | 0.29 | 0.33 | 0.38 | 0.58 | 0.29 | 0.50 | 0.38 | 0.75 | 1.00 | 0.17 | Y |
| V | 0.75 | 0.13 | 0.46 | 0.38 | 0.46 | 0.21 | 0.29 | 0.50 | 0.21 | 0.88 | 0.88 | 0.17 | 0.71 | 0.58 | 0.71 | 0.33 | 0.46 | 0.33 | 0.17 | 1.00 | V |
| | A | R | N | D | C | E | Q | G | H | I | L | K | M | F | P | S | T | W | Y | V | |

The valid substitutions, defined to have a score 0.50 and above, are shaded in grey. Table – 1 is used as the base table for the development of the proposed method. As binary numbers are user friendly, amino acid substitution group is determined using binary numbers.

**Proposed Method**

In this section we provide a method of verifying if two given protein sequences are aligned using binary string. Let X denote Table – 1 and each entry in this table be denoted by $x_{ij}$

**Construction of Binary Table**

We construct a 20 x 20 table amino acid table A as follows.

**Row:**

Each row of the table represents one of the twenty amino acids.

**Column:**

Each row of the table represents one of the twenty amino acids.

Let us denote entries of the table as aij, $1 \leq i \leq 20$, $1 \leq j \leq 20$. Each aij is a binary string of length 9 constructed as follows. Here xij denotes the $ij^{th}$ value of Table – 1. The bit constructions are seen in Table – 2.

**Bit Construction Table**

**Table – 2**

| S. No | Bits | Property | String |
|---|---|---|---|
| 1 | 1, 2, 3 | | Any binary string of length 3 |
| 2 | 4, 5, 6 | $0 \leq xij < 0.5$ | 000 |
| | | $xij \geq 0.5$ | 111 |
| 3 | 7, 8, 9 | $0 \leq xij < 0.5$ | 000 |
| | | $0.5 \leq xij < 0.7$ | 100 |
| | | $0.7 \leq xij < 0.8$ | 110 |
| | | $xij \geq 0.8$ | 111 |

**A sample table thus generated using this procedure is given in Table – 3.**

**Observe the way the bits are constructed**

- In the first three bits the maximum number of 1's possible is 3. This in done only for those entry whose xij value is atleast 0.5 For all other combinations the value is 000.
- In bit 4, 5, 6 if xij is atleast 0.5 the value is 111 else 000.
- Also in bits 7, 8, 9 the number of 1's in every bit is atleast 1 when the xij value is atleast 0.5.

**Binary matrix of substitution scores**

**Table – 3**

| | A | R | N | D | C | E | Q | G | H | I | L | K | M | F | P | S | T | W | Y | V | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A – 000 | 000111 | 000000 | 000000 | 000000 | 000111 | 000000 | 000000 | 000111 | 000000 | 000111 | 000111 | 000000 | 000111 | 000000 | 000111 | 000111 | 000000 | 000000 | 000000 | 000111 | A |
| | 111 | 000 | 000 | 000 | 100 | 000 | 000 | 110 | 000 | 110 | 110 | 000 | 110 | 000 | 100 | 100 | 000 | 000 | 000 | 110 | |
| R – 001 | 001000 | 001111 | 001000 | 001000 | 001000 | 001111 | 001111 | 001000 | 001111 | 001000 | 001000 | 001111 | 001000 | 001111 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | R |
| | 000 | 11 | 000 | 000 | 000 | 00 | 00 | 000 | 00 | 000 | 000 | 11 | 000 | 00 | 000 | 000 | 000 | 000 | 000 | 000 | |
| N – 010 | 010000 | 010000 | 010111 | 010111 | 010111 | 010111 | 010111 | 010000 | 010111 | 010000 | 010000 | 010000 | 010111 | 010000 | 010111 | 010111 | 010000 | 010000 | 010000 | 010000 | N |
| | 000 | 000 | 111 | 110 | 100 | 100 | 111 | 000 | 100 | 000 | 000 | 000 | 100 | 000 | 100 | 100 | 000 | 000 | 000 | 000 | |
| D – 011 | 011000 | 011000 | 011111 | 011111 | 011000 | 011111 | 011111 | 011000 | 011111 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | D |
| | 000 | 000 | 10 | 11 | 000 | 11 | 11 | 000 | 00 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | |
| C – 100 | 100111 | 100000 | 100111 | 100000 | 100111 | 100000 | 100000 | 100000 | 100000 | 100000 | 100000 | 100000 | 100111 | 100000 | 100000 | 100111 | 100000 | 100000 | 100000 | 100000 | C |
| | 100 | 000 | 100 | 000 | 111 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 100 | 000 | 000 | 100 | 000 | 000 | 000 | 000 | |
| E – 101 | 101000 | 101111 | 101111 | 101111 | 101000 | 101111 | 101111 | 101000 | 101111 | 101000 | 101000 | 101111 | 101000 | 101111 | 101000 | 101000 | 101000 | 101000 | 101000 | 101000 | E |
| | 000 | 00 | 00 | 11 | 000 | 11 | 10 | 000 | 00 | 000 | 000 | 00 | 000 | 00 | 000 | 000 | 000 | 000 | 000 | 000 | |
| Q – 110 | 110000 | 110111 | 110111 | 110111 | 110000 | 110111 | 110111 | 110000 | 110111 | 110000 | 110000 | 110000 | 110111 | 110111 | 110111 | 110000 | 110000 | 110000 | 110000 | 110000 | Q |
| | 000 | 00 | 00 | 00 | 000 | 00 | 11 | 000 | 00 | 000 | 000 | 000 | 00 | 00 | 00 | 000 | 000 | 000 | 000 | 000 | |
| G – 111 | 111111 | 111000 | 111000 | 111000 | 111000 | 111000 | 111000 | 111111 | 111000 | 111000 | 111000 | 111000 | 111000 | 111000 | 111000 | 111111 | 111111 | 111000 | 111111 | 111111 | G |
| | 10 | 000 | 000 | 000 | 000 | 000 | 000 | 11 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 11 | 11 | 000 | 00 | 00 | |
| H – 000 | 000000 | 000111 | 000111 | 000111 | 000000 | 000111 | 000111 | 000000 | 000111 | 000000 | 000000 | 000111 | 000000 | 000000 | 000000 | 000111 | 000000 | 000000 | 000111 | 000000 | H |
| | 000 | 100 | 100 | 100 | 000 | 100 | 100 | 000 | 111 | 000 | 000 | 100 | 000 | 000 | 000 | 110 | 000 | 000 | 100 | 000 | |
| I – 001 | 001111 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001111 | 001111 | 001000 | 001111 | 001111 | 001111 | 001000 | 001000 | 001111 | 001111 | 001000 | I |
| | 10 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 11 | 11 | 000 | 10 | 00 | 00 | 000 | 000 | 00 | 10 | 000 | |
| L – 010 | 010111 | 010000 | 010000 | 010000 | 010111 | 010000 | 010000 | 010111 | 010000 | 010111 | 010111 | 010000 | 010111 | 010111 | 010111 | 010000 | 010000 | 010000 | 010111 | 010000 | L |
| | 110 | 000 | 000 | 000 | 100 | 000 | 000 | 100 | 000 | 111 | 111 | 000 | 111 | 110 | 100 | 000 | 000 | 000 | 111 | 000 | |
| K – 011 | 011000 | 011111 | 011000 | 011000 | 011000 | 011111 | 011111 | 011000 | 011111 | 011000 | 011000 | 011111 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | K |
| | 000 | 11 | 000 | 000 | 000 | 00 | 00 | 000 | 00 | 000 | 000 | 11 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | |
| M – 100 | 100111 | 100000 | 100111 | 100000 | 100111 | 100000 | 100000 | 100111 | 100000 | 100111 | 100111 | 100000 | 100111 | 100111 | 100111 | 100000 | 100000 | 100111 | 100000 | 100111 | M |
| | 110 | 000 | 100 | 000 | 100 | 000 | 000 | 100 | 000 | 110 | 111 | 000 | 111 | 110 | 100 | 000 | 000 | 100 | 000 | 110 | |
| F – 101 | 101111 | 101000 | 101000 | 101000 | 101111 | 101000 | 101000 | 101000 | 101000 | 101111 | 101111 | 101000 | 101111 | 101111 | 101111 | 101000 | 101000 | 101111 | 101111 | 101111 | F |
| | 00 | 000 | 000 | 000 | 00 | 000 | 000 | 000 | 000 | 00 | 10 | 000 | 10 | 11 | 00 | 000 | 000 | 00 | 00 | 00 | |
| P – 110 | 110111 | 110000 | 110000 | 110000 | 110000 | 110000 | 110000 | 110111 | 110000 | 110111 | 110111 | 110000 | 110111 | 110111 | 110111 | 110000 | 110000 | 110111 | 110000 | 110111 | P |
| | 00 | 000 | 000 | 000 | 000 | 000 | 000 | 00 | 000 | 00 | 00 | 000 | 00 | 00 | 11 | 000 | 000 | 00 | 000 | 10 | |
| S – 111 | 111111 | 111000 | 111111 | 111000 | 111000 | 111000 | 111111 | 111111 | 111000 | 111000 | 111000 | 111000 | 111000 | 111000 | 111000 | 111111 | 111111 | 111000 | 111111 | 111000 | S |
| | 00 | 000 | 00 | 000 | 00 | 000 | 00 | 00 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 11 | 10 | 000 | 00 | 000 | |
| T – 000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 | 000111 | 000111 | 000000 | 000000 | 000000 | T |
| | 000 | 000 | 100 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 110 | 111 | 000 | 000 | 000 | |
| W – 001 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001000 | 001111 | 001111 | 001000 | W |
| | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 11 | 10 | 000 | |
| Y – 010 | 010000 | 010000 | 010000 | 010000 | 010000 | 010000 | 010000 | 010000 | 010000 | 010111 | 010111 | 010000 | 010000 | 010111 | 010000 | 010111 | 010000 | 010111 | 010111 | 010000 | Y |
| | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 000 | 100 | 000 | 100 | 000 | 110 | 111 | 000 | |
| V – 011 | 011111 | 011000 | 011000 | 011000 | 011000 | 011000 | 011000 | 011111 | 011000 | 011111 | 011111 | 011000 | 011111 | 011111 | 011111 | 011000 | 011000 | 011000 | 011000 | 011111 | V |
| | 10 | 000 | 000 | 000 | 000 | 000 | 000 | 00 | 000 | 11 | 11 | 000 | 10 | 00 | 10 | 000 | 000 | 000 | 000 | 11 | |
| | A | R | N | D | C | E | Q | G | H | I | L | K | M | F | P | S | T | W | Y | V | |

So note that the table has been suitably constructed so that for any entry in Table – 1 whose value is selected as a amino acid substitution group, the number of nonzero entry in Table – 2 is atleast 4. We use this property for the protein sequence alignment.

**Proposed Protein Sequence Alignment Method**

When two protein sequences are given the regular alignment is the sequences match with each other exactly the same. In cases that is not possible we try to match two sequences when atleast half of the properties match. From Table – 1 protein A and proteins C, G, I, L, M, F, P, S, V are similar in sense that they share atleast 0. 5 of the properties. This means that in protein sequence alignment the pairs ( A, C ), ( A, G ), ( A, I ), ( A, L ), ( A, M ), ( A, F ), ( A, P ), ( A, S ) are considered to be aligned. Based on this discussion we propose a method for determining if two sequences are aligned with atleast 50% properties satisfied.

Let S1: m1 m2 m3… mk and S2: n1 n2 … nk be the protein sequences of length k to be verified for alignment.

**Step 1** Choose the pairs ( m1 n1 ), ( m2 n2 ) … ( mk nk ).
**Step 2** Assign the binary value from Table – 2 where m1, m2,…mk represents the corresponding rows and n1, n2,…, nk represents the corresponding columns of Table – 3
**Step 3** For each binary segment of length p = 9 we count the number of non zero entries to generate a sequence M.
**Step 4** If all the entries in the sequence M are ≥ 4, then the two sequences are aligned. Else if atleast one value is < 4, then the sequences are not aligned.

For example if S1: A D L K M V Y and S2: R E G K F P Y be the sequences to be aligned. We construct the sample Table – 4 based on the algorithm.

In the second example it can be seen that in normal sequence alignment sense they do not match even in one position. But they match in sense that they match by satisfying the condition that atleast half of the properties match.

**Sample Table**

**Table – 4**

| S1 | A | D | L | K | M | V | Y |
|---|---|---|---|---|---|---|---|
| S2 | R | E | G | K | F | P | Y |
| Binary String | 000000 000 | 011111 111 | 010111 100 | 011111 111 | 100111 110 | 011111 110 | 010111 111 |
| No of nonzero entry | 0 | 8 | 5 | 8 | 6 | 7 | 7 |
| Conclusion | **The sequences are not aligned since not entries are ≥ 4** | | | | | | |

| S1 | I | M | D | G | T | Y | V |
|---|---|---|---|---|---|---|---|
| S2 | M | P | H | L | S | H | M |
| Binary String | 001111 110 | 100111 100 | 011111 110 | 111111 100 | 111111 111 | 0101111 00 | 011111 110 |
| No of nonzero entry | 6 | 5 | 7 | 7 | 9 | 5 | 7 |
| Conclusion | **The sequences are aligned since all entries are ≥ 4** | | | | | | |

Higher the decimal values in the third row of the example the better the properties shared by the sequences even if they do not match with each other. This is made possible by the values assigned to the last three bits in Table – 3 , 111 is assigned only if more than 80% of the properties are same. So larger the decimal values, better the alignment.

## Conclusion

In the proposed method

- The sequence used for alignment are binary strings and hence computation friendly.
- The verification procedure is not difficult and hence user friendly and can be programmed easy.
- This can be used for regular protein sequence alignment and also used to verify if the sequences satisfy atleast 50% of the properties.

So the proposed method is user friendly and can be used for protein sequence alignment. Also this method can be used for encrypting details regarding protein sequences. Any protein sequence can be converted into binary strings using Table and hence can be encrypted. Many binary strings are available in public domain and hence safe for encryption. So the proposed method is safe and compactable for sequence verification and protein sequence encryption.

## References

1. http://scholarworks.gsu.edu/cs_diss
2. Shailendra Singh , Multiple Sequence Alignment using Boolean Algebra and Fuzzy Logic: A Comparative Study Nivit Gill, Int. J. Comp. Tech. Appl., Vol 2 (5), 1145-1152.
3. http://arxiv.org/ftp/arxiv/papers/1208/1208.5713.pdf.
4. http://en.wikipedia.org/wiki/Sequence alignment.
5. biochem.stanford.edu/biochem218/Projects%202001/Yu.pdf.

*****