**Course: Advanced Bioinformatics**

**Module title: Distance Measures**

**Module no. : 27**

In this module, detail about distance measured is mentioned that how can we compare two sequences or two strings? Well there are several methods; in this module we will cover the very basic ones.

**Hamming or edit distance:** One method in determining sequence similarity is to determine the edit distance between two sequences. If we take the example of pear and tear, how similar are these two words? We notice that if we change the p to a t, and keep the ear, then we can change pear to tear. Thus, there is a mismatch in the first letter, and matches in the last three. An alignment of these two is as follows:

**B E A R**

**| | |**

**N E A R**

One way to score this alignment is to calculate the ***Hamming distance***, which is the minimum number of letters by which the two words differ. In this example, the Hamming distance would be 1. The Hamming distance is calculated by summing up the number of mismatches when two words are aligned to one another.

With biological sequences, it is often necessary to align two sequences that are of different lengths, or that have regions that have been inserted or deleted over time. Thus, the notion of gaps needs to be introduced. Consider the words **alignment** and **ligament**. One alignment of these two words is as follows:

**A L I G N M E N T**

**| | | | | | |**

**- L I G A M E N T**

In this case, a gap is denoted in the alignment by a ‘-‘character. Now an alignment can produce one of the following: a match between two characters, a mismatch between two characters (also called a substitution or mutation), a gap in the first sequence (which can be thought of as the deletion of a character in the first sequence), or a gap in the second sequence (which can be thought of as the insertion of a character in the first sequence).

Consider the following two nucleic acid sequences: **ACGGACT** and **ATCGGATCT**. The following are two valid alignments:

**A – C – G G – A C T**

**| | | | |**

**A T C G G A T \_ C T**

**A T C G G A T C T**

**| | | | | |**

**A – C G G – A C T**

Longest Common Subsequence

x=ABCBDAB and

y=BDCABA,

BCA is a common subsequence and

BCBA and BDAB are two LCSs

N- grams Methods: Subdivide words into N- grams -set of overlapping

substrings of length N

N=2:

(radio) (ra - ad - di -io)

N=3:

(radio) (rad - adi - dio)

**Alignment scoring schemes**

Which alignment is the better alignment? One way to judge this is to assign a positive score for each match, and a negative score for each mismatch, and a negative score for each insertion/deletion (collectively referred to as indels).

One scoring scheme might assign the following values:

match: +2

mismatch: -1

indel –2

Using this scoring scheme, the first alignment has 5 matches, 1 mismatch, and 4 indels. The score for this alignment is: 5 \* 2 – 1(1) – 4(2) = 10 – 1 – 8 = 1.

The second alignment has 6 matches, 1 mismatch, and 2 indels. The score for the second alignment is 6 \* 2 – 1(1) – 2 (2) = 12 – 1 – 4 = 7.

Therefore, using the above scoring scheme, the second alignment is a better alignment, since it produces a higher alignment score.