**Course: Advanced Bioinformatics**

**Module title: MSA Approaches**

**Module no. : 38**

In this module, we’ll see that what is the need of MSA and where it is used in the area of Bioinformatics?

Similar genes can be conserved across species that perform similar or identical functions. Many genes are represented in highly conserved forms across organisms. By performing a simultaneous alignment of multiple sequences having similar or identical functions, we can gain information about which regions have been subject to mutations over evolutionary time and which are evolutionarily conserved. Such knowledge tells which regions or domains of a gene are critical to its functionality.

Sometimes genes that are similar in sequence can be mutated or rearranged to perform an altered function. By looking at multiple alignments of such sequences, we can tell which changes in the sequence have caused a change in the functionality.

Multiple sequence alignment yields information concerning the structure and function of proteins, and can help lead to the discovery of important sequence domains or motifs with biological significance while at the same time uncovering evolutionary relationships among genes.

In multiple sequence alignment, the idea is to take three or more sequences, and align them so that the greatest number of similar characters are aligned in the same column of the alignment.

The difficulty with multiple sequence alignment is that now there are a number of different combinations of matches, insertions, and deletions that must be considered when looking at several different sequences. Methods to guarantee the highest scoring alignment are not feasible. Therefore, approximation methods are put to use in multiple sequence alignment.

Here we give a list of MSA algorithms. Broadly we can divide MSA algorithms into six categories.

1. **MSA Basic Approach:** Hierarchical extensions of pairwise alignment

Principal: successive application of pairwise methods.

MSA Procedure

* + 1. For four sequences S1 ,..S4
    2. 6 pairwise comparisons,

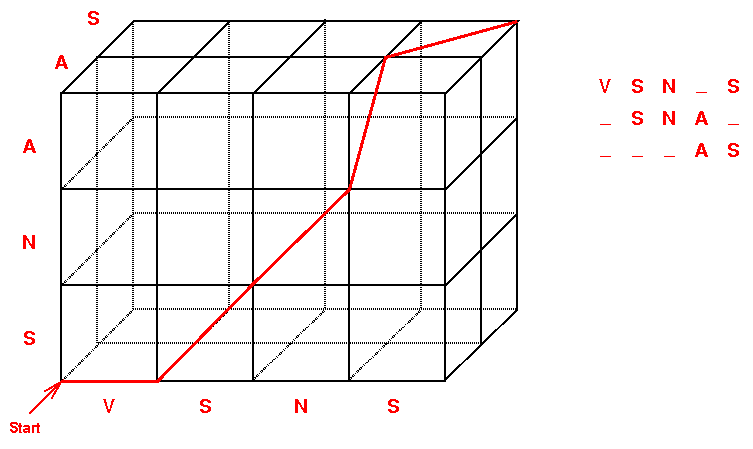
Optimal vs correct: No single "correct" alignment. “optimal" w.r.t set of calculations. This is partly due to: the complexity of the problem, limitations of the scoring systems used and our limited understanding of life and evolution.

1. **Exhaustive search:** extension of DP to multi-dimensions.
2. **Progressive alignment:** compute tree of sequences, based on hierarchical clustering, and then merge closest first, greedily.
3. **Conserved blocks:** find highly conserved regions and then grow alignment around these regions. e.g. BLAST
4. **Iterative search:** based on genetic algorithm
5. **Probabilistic/statistical:**
   * 1. Gibbs Sampling, HMM

### Extension of Dynamic Programming Approach

The attractiveness of dynamic programming with two sequences is that it guarantees to give the optimal alignment of sequences given a specific scoring scheme. In addition, it is a relatively easy method to implement.

Dynamic programming approaches can be extended to multiple alignment as well. Consider the example where we have three amino acid sequences VSNS, SNA, and AS to align. Instead of filling a two dimensional matrix as we did with two sequences, we now fill a three dimensional space.



Suppose the length of each sequence is n residues. If there are two such sequences, then the number of comparisons needed to fill in the scoring matrix is n2, since it is a two-dimensional matrix. The number of comparisons needed to fill in the scoring cube when three sequences are aligned is n3, and when four sequences are aligned, the number of comparisons needed is n4. Thus, as the number of sequences increases, the number of comparisons needed increases exponentially, i.e. nN where n is the length of the sequences, and N is the number of sequences. Thus, without any changes to the dynamic programming approach, this becomes impractical for even a small number of short sequences rather quickly.