**Course: Advance Bio Informatics**

**Module Title: Fitch / Margoliash Algorithm**

**Module No: 64**

The Fitch-Margoliash method uses a weighted least squares method for clustering based on genetic distance. Closely related sequences are given more weight in the tree construction process to correct for the increased inaccuracy in measuring distances between distantly related sequences. The distances used as input to the algorithm must be normalized to prevent large artifacts in computing relationships between closely related and distantly related groups. The distances calculated by this method must be linear; the linearity criterion for distances requires that the expected values of the branch lengths for two individual branches must equal the expected value of the sum of the two branch distances - a property that applies to biological sequences only when they have been corrected for the possibility of back mutations at individual sites.

This correction is done through the use of a substitution matrix such as that derived from the Jukes-Cantor model of DNA evolution. The distance correction is only necessary in practice when the evolution rates differ among branches. Another modification of the algorithm can be helpful, especially in case of concentrated distances (please report to concentration of measure phenomenon and curse of dimensionality): that modification, described in, has been shown to improve the efficiency of the algorithm and its robustness.

The least-squares criterion applied to these distances is more accurate but less efficient than the neighbor-joining methods. An additional improvement that corrects for correlations between distances that arise from many closely related sequences in the data set can also be applied at increased computational cost. Finding the optimal least-squares tree with any correction factor is NP-complete, so heuristic search methods like those used in maximum-parsimony analysis are applied to the search through tree space.

Following are the steps for **Fitch Algorithm**.

1. Determine Length of given tree
2. Find out root of tree
3. Find out internal nodes of tree
4. Root the tree at an arbitrary internal node (or internal branch)
5. Visit an internal node x for which no state set has been defined, but where the state sets of x's immediate descendants (y,z) have been defined.
6. If the state sets of y,z have common states, then assign these to x.
7. If there are no common states, then assign the union of y,z to x, and increase tree length by one.
8. Repeat until all internal nodes have been visited.
9. Note length of current tree.

