**Course: Advanced Bioinformatics**

**Module title: Phylogenetic Analysis**

**Module no. : 212**

You can use functions for phylogenetic tree building and analysis. There is also a GUI to draw phylograms (trees).

**Phylogenetic tree data —** Read and write Newick-formatted tree files (phytreeread, phytreewrite) into the MATLAB Workspace as phylogenetic tree objects (phytree).

**Create a phylogenetic tree —** Calculate the pairwise distance between biological sequences (seqpdist), estimate the substitution rates (dnds, dndsml), build a phylogenetic tree from pairwise distances (seqlinkage, seqneighjoin, reroot), and view the tree in an interactive GUI that allows you to view, edit, and explore the data (phytreeviewer or view). This GUI also allows you to prune branches, reorder, rename, and explore distances.

**Phylogenetic tree object methods —** You can access the functionality of the phytreeviewer GUI using methods for a phylogenetic tree object (phytree). Get property values (get) and node names (getbyname). Calculate the patristic distances between pairs of leaf nodes (pdist, weights) and draw a phylogenetic tree object in a MATLAB Figure window as a phylogram, cladogram, or radial treeplot (plot). Manipulate tree data by selecting branches and leaves using a specified criterion (select, subtree) and removing nodes (prune). Compare trees (getcanonical) and use Newick-formatted strings (getnewickstr).

**Microarray Data Analysis**

The MATLAB environment is widely used for microarray data analysis, including reading, filtering, normalizing, and visualizing microarray data. However, the standard normalization and visualization tools that scientists use can be difficult to implement. The toolbox includes these standard functions:

**Microarray data —** Read Affymetrix GeneChip files (affyread) and plot data (probesetplot), ImaGene results files (imageneread), SPOT files (sptread) and Agilent microarray scanner files (agferead). Read GenePix GPR files (gprread) and GAL files (galread). Get Gene Expression Omnibus (GEO) data from the Web (getgeodata) and read GEO data from files (geosoftread). A utility function (magetfield) extracts data from one of the microarray reader functions (gprread, agferead, sptread, imageneread).

**Microarray normalization and filtering —** The toolbox provides a number of methods for normalizing microarray data, such as lowess normalization (malowess) and mean normalization (manorm), or across multiple arrays (quantilenorm). You can use filtering functions to clean raw data before analysis (geneentropyfilter, genelowvalfilter, generangefilter, genevarfilter), and calculate the range and variance of values (exprprofrange, exprprofvar).

**Microarray visualization —** The toolbox contains routines for visualizing microarray data. These routines include spatial plots of microarray data (maimage, redgreencmap), box plots (maboxplot), loglog plots (maloglog), and intensity-ratio plots (mairplot). You can also view clustered expression profiles (clustergram, redgreencmap). You can create 2-D scatter plots of principal components from the microarray data (mapcaplot).

**Microarray Data Storage** The toolbox includes functions, objects, and methods for creating, storing, and accessing microarray data.

The object constructor function, DataMatrix, lets you create a DataMatrix object to encapsulate data and metadata from a microarray experiment. A DataMatrix object stores experimental data in a matrix, with rows typically corresponding to gene names or probe identifiers, and columns typically corresponding to sample identifiers. A DataMatrix object also stores metadata, including the gene names or probe identifiers (as the row names) and sample identifiers (as the column names).

You can reference microarray expression values in a DataMatrix object the same way you reference data in a MATLAB array, that is, by using linear or logical indexing. Alternately, you can reference this experimental data by gene (probe) identifiers and sample identifiers. Indexing by these identifiers lets you quickly and conveniently access subsets of the data without having to maintain additional index arrays.

Many MATLAB operators and arithmetic functions are available to DataMatrix objects by means of methods. These methods let you modify, combine, compare, analyze, plot, and access information from DataMatrix objects. Additionally, you can easily extend the functionality by using general element-wise functions, dmarrayfun and dmbsxfun, and by manually accessing the properties of a DataMatrix object.