**Course: Advanced Bioinformatics**

**Module title: Matlab Introduction**

**Module no. : 207**

**Matlab and its basics:**

Matlab stands for Matrix Laboratory which means that all the programming done is in the form of matrices which is a common concept in mathematics.

Matlab is based on Mathematical Programs. It is widely used in Education & Industry. Programs written in Matlab range from simple algebra to complex Calculus. Beauty of matlab is that programs written in mantlab can be converted to C, C++ etc. Further, matlab can also be using in high level languages such as using ActiveX Object for

VB, and VC etc.

Operating Matlab Window:

To Start:

Windows platform select

Start->Programs->MATLAB version no

MATLAB also runs on Linux, Unix and Solaris and other platforms.

To stop (nicely)

Select File -> Exit MATLAB or type quit in the MATLAB command window

When we open matlab, there are three major portions of matlab screen.

Command Window is the place where we write commands. Workspace presents information about variables generated based on the commands. Command history shows list of commands used in our Matlab software.

You'll notice the three main windows are "Workspace", "Command History", and "Command Window".

The Workspace window is a list of everything used currently in your workspace.  In this case it contains three matrices or arrays of size 1 X 80000.  You'll learn more about what can appear here later.

The Command History window shows the list of commands that you've executed already.

The Command Window window is probably the one that you'll spend the most time looking at (which is why it's the biggest).  It shows results, commands, errors, etc...  Basically, everything you'd want to know about what's currently going on.

The next part of the tutorial comes directly from the script file above.  Please use that if available to follow along, this is simply a more printable version of the tutorial.  
  
**Assignments**

Here are a few basic assignment statements think of an assignment as a variable.  
  
a = 25;  
b = 42\*12;  
c = 50  
d = 36\*12 + 120  
  
Note that when the semicolon is present that Matlab doesn't display the results on the screen, otherwise is does display the results after the command is executed.  
  
Important commands  
clear command  
The clear command is very important, it simply clears a variable from the workspace.  
  
clear a;  
clear b  
clear c  
clear all;  
  
"clear all" clears EVERYTHING from the Matlab Workspace.  It is recommended that you always put "clear all" as the first line of your M-file so that you don't have anything in your workspace that may cause problems with your script.  
  
help command  
The help command is another critical command to know. If you don't know what a particular command does you can always type "help <command>" and it will display what the command does and how to use it.  
  
help clear;  
help help;  
  
lookfor command  
The lookfor command is for that time when you don't know what the name of the command you're looking for is. Type "lookfor <keyword>" and it will search the first line of the comments for each command and return matches. Note, it sometime can take awhile.  
  
lookfor fourier;  
  
**Vectors**

Vectors form the basis of everything in Matlab. Becoming familiar with this concept is a must when working with Matlab. Think of a vector as a 1 X N matrix. Since Matlab is a numerical solver it requires that input to Matlab be completely numerical (not algebraic).  
  
a = [1 2 3]  
b = [4 5 6]  
  
a  
b  
  
a\*b  
This command returned an error didn't it? This is because Matlab tried to multiply a 1 X 3 vector times a 1 X 3 vector. Since vectors are really just 1 X N matrices this obviously creates a problem. You need to first decide what you desire from the result. Do you want the two to be multiplied like two matrices or do you want the corresponding elements of each matrix multiplied individually?  
  
The tick mark ' (same key as double quote) is used to imply the transpose of a matrix. Note, the transpose is different from the inverse which is the inv(<variable>) command. So if you desired to multiply a\*b like two matrices then you need to transpose the second (or first) matrix.  
a\*b'  
a'\*b  
b'\*a  
b\*a'  
remember matrix multiplication is noncommutative so var1\*var2 is not necessarily equal to var2\*var1.  
  
If instead you wanted to have each corresponding element of a and b multiplied to form another 1 X 3 matrix (in this case) then you need to use Matlabs "dot notation". The dot notation assumes that you are running the command element by element.  
  
a.\*b  
a./b  
  
Notice how this command multiplied (or divided) 1\*4, then 2\*5, then 3\*6? Often times you will forget to put the period in your equations, but it will usually give you an error (but not always!).