MATLAB Good Programming Practices

Harvard Systems Biology
Summer 2009

Mike Springer **michael_springer@hms.harvard.edu**
Zeba Wunderlich **zeba@hms.harvard.edu**

1. *Use variables instead of hard coded numbers. Put these numbers at the top of your scripts and functions.*
   Say you there's a number you use in your code – e.g. you want to look at every third element in your vector **`x`**. Instead of **`y = 1:3:end`**, put the line **`intervalSize = 3`** at the top of your script and change the line to **`y = 1:intervalSize:end`**. Want to change the interval size? You only have to change it in one (easy-to-find) place.

2. *Write functions for things you do over and over again.*
   This follows pretty much the same logic as the point above. Need to change how you calculate something? Change it in one place instead of all over your code. You can make a special folder for your own MATLAB functions and add it to your path to make life easier.

3. *Use descriptive variable names.*
   As useful as **`x`** and **`y`** are as variable names, when you look back at your script a month later, you probably will have no idea what they mean.

4. *Put in comments to describe tricky parts of your code.*
   It happens to everyone. You find a really clever, one-line solution to your problem. Your write it. It works. You go back to it at a later date and wonder what the heck your clever, one-line solution does. Explain the solution in a comment. Feel good about yourself every time you read the comment.

5. *Document your functions.*
   Along the same lines as the point above, document how your functions work. You know the stuff that's printed out when you type **`help <function>.`** This can work for your functions, too! All the comments between your function definition and the first line of code will print out when you type **`help <function>.`** More info can be found here:
   http://www.mathworks.com/access/helpdesk/help/techdoc/matlab_prog/f7-41453.html#f7-38710

6. *Use MATLAB built-in functions when they are available.*
   Are you thinking to yourself, "MATLAB must have a function for this!"? It probably does. To find it, try:

   a. If you know a similar function, type **`help <similar function>.`** At the bottom, there is a list of related functions. See if the one you are looking for is there.

b. Try searching the online help:
   http://www.mathworks.com/access/helpdesk/help/helpdesk.html
c. Google it! (Bing it?)
d. Ask a friend who uses MATLAB a lot.

7. *Learn how to use structures and cell arrays well.*
   Good data structures will save you time and aggravation.

8. *Write functions to input and output data file types you often use.*

9. *Sanity check your code.*
   Try simple examples to make sure it works properly before trying it on a big data set.

10. *Learn how to use the MATLAB debugger functions.*
    See the documentation here:
    http://www.mathworks.com/access/helpdesk/help/techdoc/matlab_prog/f10-60570.html

11. *Write scripts for each figure you need to make.*
    MATLAB's plotting defaults are pretty ugly, so to make publication- or presentation-quality figures requires some tweaking. There are a variety of commands to do this. Below is a snippet of a script to make a figure, along with tweaks to line widths, font sizes, fonts, etc.

More programming tips are available here:
http://www.mathworks.com/access/helpdesk/help/techdoc/matlab_prog/f10-57434.html

```
load infoContent-R.txt
load infoContent-J.txt
load infoContent-Y.txt

lightBlue = [100/256, 149/256, 237/256];
darkBlue = [0/256, 0/256, 128/256];
slateGray = [198/256 226/256, 255/256];

fontSize = 16;
fontName = 'Times';

figure;
set(gcf, 'Units', 'inches', 'Position', [4, 1, 15, 4])
subplot(1,3,1)

b = infoContent_R(:,2);
y = infoContent_Y(:,2);
e = infoContent_J(:,2);

minEdge = floor(min([min(b) min(y) min(e)]));
maxEdge = ceil(max([max(b) max(y) max(e)]));
edges = minEdge:4:maxEdge;

nB = histc(b,edges)/length(b);
nY = histc(y,edges)/length(y);
nE = histc(e,edges)/length(e);
```

```
n = [nB nY nE];
h = bar(edges, n, 1);

set(h(1),'FaceColor',darkBlue)
set(h(2),'FaceColor',lightBlue)
set(h(3),'FaceColor',slateGray)

xlim([0 maxEdge])
set(gca, 'FontSize', fontSize, 'FontName', fontName, 'LineWidth', 2);
set(gca, 'XTick', 0:20:maxEdge);
xlabel('L, bps')
ylabel('frequency')
```