Matlab tips [x.chen1@ncl.ac.uk](mailto:x.chen1@ncl.ac.uk) Xing

Write function name, author name, date, and helpful description of your script at the top of your code. If you type in ‘help ‘ followed by the name of the function at the prompt, Matlab will print the description in the first block of text (everything that is commented up to the first empty line).

>> help example\_header\_code

Keep a list of your important functions somewhere, with descriptions. You could use Microsoft Word’s ‘styles’ to easily differentiate code from the rest of the text, e.g. in this document, lines of code are set to the style ‘code’ while descriptive text is set to ‘description.’

Comment code using the ‘%’ sign. Keyboard shortcut: Ctrl+R. To uncomment: Ctrl+T.

Heed conventions to facilitate code sharing & enhance readability (refer to johnson\_Matlab\_style\_guide.pdf). E.g. Capitalise first letter of all words after the first; use all caps only for GLOBAL variables.

Initialise variables when you first use them, to avoid mistakes later.

initialise\_variables

Check whether a variable exists, type of variable.

exist(‘var1’)

As far as possible, minimise ‘hard coding’ and use variables instead.

minimise\_hard\_coding

Keep indentation neat with Ctrl+I.

Open highlighted function with Ctrl+D.

Press F9 to run highlighted code, or to print a highlighted variable to the screen.

At the prompt, use the ‘Up’ arrow key to cycle through lines of code that were previously run at the prompt, for easy access. Or drag a command from the Command History window to the Command Window.

Type in first letters of commands in the Command History window to locate commands that match these letters.

When debugging, press F5 to continue.

Use meaningful folder and file names. Be descriptive. Anticipate the explosion of collected data, iterations of analyses, and proliferation of folders. Create subfolders where needed. Name folders using variables names and numbers where needed. Incorporate date/time stamps if that will help.

In Matlab Window, use Edit>Find or Ctrl+Shift+F to find files based on their names or to search through their text for key words.

Use the split screen tool in the Editor Window to browse, copy and paste easily between scripts.

Use Tools>Compare Against to compare two pieces of code against each other (good for the initial parts of the code, but gets lost after a lot of differences accumulate).

Saving data in ‘.mat’ format to a particular folder. How to set the full path to the folder ‘automatically’ rather than manually; how to make directories automatically if they don’t already exist; how to check whether a file of the same name already exists; how to save and/or overwrite a mat file. An example here:

save\_mat\_file.m

You can use Matlab commands to create Excel spreadsheets, write and read data, and retrieve info about particular spreadsheets.

excel\_read\_write

Other handy tips:

MS Word: Press Shift+F5 multiple times, to cycle cursor across its last three to four locations in a document. Very handy for editing long documents.

Doubleclick a word to highlight it. Triple click to highlight a line (in Matlab) or a paragraph (in MS Word).

Use the Ctrl key to move the cursor word by word, rather than letter by letter. Home and End move the cursor to the start and end of it current line. Ctrl+Home moves the cursor to the top of the text, while Ctrl+End moves it to the bottom.

Alt+Tab cycles between windows (I think it’s Command+Tab on a Mac). Hold down the Alt key and press the Tab key repeatedly. To cycle through in the opposite direction, hold down ALT and Shift at the same time, while pressing Tab.

Obtain a clear view of your desktop in Windows with ‘Windows key’+D.

Next topics:

5/6/14

Use cells for data arrays of varying length

Use Help function

Create account on Mathworks site to post and answer questions

For overnight runs, use error throwing and catching to process data wherever possible and skip errors that arise unexpectedly with parts of dataset

Use a version control system like Git, to keep code organised; facilitate retrieval; find code that was created at a similar period; recall code’s purpose; and potentially to share code with collaborators

12/6/14

Use Autohotkey to automate repetitive tasks (file management, routine analyses)

How to implement commonly used statistics in Matlab: *t*-tests, correlations, ANOVAs, signed rank/ rank sum, chi-squared test. More advanced: circular stats, regression, goodness of fit, skewness, Levene’s test, Akaike information criterion, partial correlation.

19/6/14

How to export figures, choose image formats, use transparency, create publication-quality figures, use a standard template of image settings.

Subject to demand:

Use parallel toolbox for large datasets and high throughput processing