How to write good code Part 2

Content

Part 1:

- Naming Conventions
- Code Layout
- Commenting

Part 2:

- Functionising
- Conceptualisation
- Externalisation

Part 3:

- Debugging
- Testing

Why write good code?

- ‘Good code is a present for your future self’

Example code

```
%Todo: Automatically make locomotion file - DONE
% Fix fluctuation detection locomotion
% Automatically select cells that are place cells  - DONE
% Overlay ROIs on image
% Visualize results
% Make masterfile a function with variables.

% folder = 'C:\Users\dorieke\Dropbox (Brain Energy Lab)\Everything\Dori\Data\Simon\2017_11_10\20171110_13_49_05_Env1_1526Hz';
% folder = 'C:\Users\dorieke\Dropbox (Brain Energy Lab)\Everything\Dori\Data\Maggie\2017_09_15\20170915_15_24_35_PC1_12Hz';
folder = cd;
% folder = 'C:\Users\User\Dropbox (Brain Energy Lab)\Everything\Dori\Data\Simon\2017_11_10\20171110_13_55_05_Env2_1526Hz';
% placetiff = 'placecellsCropSub.tif';
placetiff = 'PC_comb.tif';
cd(folder);

    fn = fullfile(folder, placetiff);
    info = imfinfo(fn);
    num_images = numel(info);
    flims = [1,num_images];
    [mixedsig, mixedfilters, CovEvals, covtrace, movm, movtm] = CellsortPCA(fn, flims);
    % size(mixedsig)
%     [PCuse] = CellsortChoosePCs(fn, mixedfilters) ;
PCuse = [1:300];
```

What we discussed last week

- Naming conventions
- Code Layout
  - White space
  - Indenting
  - Paragraphs
  - Character limit line
- Commenting
  - Inform reading of aim
  - Variables
  - ‘If out’

Use of functions

- Compartmentalize
- Easier to share
- Easier to reuse
- Reduce workspace clutter
  - Remember scope!
- Make an API

A	*10x1 struct*
allRois	*121x123 double*
ans	'cellsort_preprocessed_data'
arealims	15
cell_sig	*214x20000 double*
CovEvals	*300x1 double*
covtrace	0.0667
dataNorm	*214x20000 double*
deconvtau	0
dirs	*1x5 cell*
dt	1
flims	[1,20000]
fn	'D:\Dropbox (Brain Energy Lab)\Everything\Dori\Data\Raw\P
folder	'D:\Dropbox (Brain Energy Lab)\Everything\Dori\Data\Raw\P
i	4
ica_A	*300x300 double*
ica_filters	*300x118x119 double*
ica_segments	*214x118x119 double*
ica_sig	*300x20000 double*
info	*20000x1 struct*
les	[1,5001,10001,15001,20001]
mixedfilters	*118x119x300 double*
mixedsig	*300x20000 double*
movm	*118x119 double*
movtm	*1x20000 double*
mu	0.2000
myDir	*10x1 logical*
normalization	1
num_images	20000
numiter	100
PCuse	*1x300 double*
placetiff	'PC_comb.tif'
segcentroid	*214x2 double*
segmentlabel	*[]*
smwidth	3
spc	*747360x1 double*
spmat	*20000x300 sparse logical*
spt	*747360x1 double*
thresh	0.5000
traces	*214x5000 double*

- Make a function that takes folder/file as input
- Only output those four variables

Code as function

```
function [dataNorm, ica_segments, segcentroid, spmat] = ...
    cellSortAnalysis(folder, placetiff)
% Aim:      Extract ROIs from a preprocessed image and determine the
%           calcium traces per ROI
% Input:    Tiff-file containing preprocessed stack of calcium imaging
% Output:   MAT-file containing the deltaF/F calcium traces for each ROI
```

Running it

```
folder   =  '.\Andi\2018_03_07'; % folder containing raw tiff file
tiffname =  'PC_comb.tif'; % tiff image of calcium recording

% Run cellsort
[dataNorm, ica_segments, segcentroid, spmat] = ...
    cellSortAnalysis(folder, tiffname);
```

Workspace after

dataNorm	*18x4000 double*
folder	'.\Andi\2018_03_07'
ica_segments	*18x128x256 double*
segcentroid	*18x2 double*
spmat	*4000x300 sparse logical*
tiffname	'PC_comb.tif'

Signs of poor conceptualisation

- Copy and paste
- Numbers in variable names
- Using eval (matlab)
- Using clear/clear all

Improve this code:

```
string01="Tick if you want to save open and save images";
Dialog.create("Analysis Options");
Dialog.addCheckbox(string01, true);
string02="Tick if you're analysing for the first time";
string03="Tick if you want to analyse microglia";
//string04="Tick if you want to analyse pericytes";
//string05="Tick if you want to add up microglia results tables";
//string03="Tick if you want to add up pericyte counts results tables";
//string03="Tick if you want to add up pericyte ISD results tables";
Dialog.addCheckbox(string02, true);
Dialog.addCheckbox(string03, true);
//Dialog.addCheckbox(string04, true); include when pericyte is done
Dialog.show() ;

Saveimages=Dialog.getCheckbox();
firsttime=Dialog.getCheckbox();
microglia=Dialog.getCheckbox();
//pericytes=Dialog.getCheckbox();

```

Improved code

```
/* Macro to analyse microglia*/

// Create dialog box for user to select options
strings = newArray( "Tick if you want to save open and save images",
                    "Tick if you're analysing for the first time",
                    "Tick if you want to analyse microglia");

Dialog.create("Analysis Options");
for (i=0; i<strings.length; i++) {
    Dialog.addCheckbox(strings[i], true);
};
Dialog.show() ;

// Get the options selected by the user
optionSelected = newArray(0,0,0);
for (i=0; i<strings.length; i++) {
    optionSelected[i] = Dialog.getCheckbox();
}
```

Signs of poor externalisation

- Using ‘cd’ in your code
  - Will not work when run on other computers
- Hardcoding numbers that are used more than once
  - Put into variable
- Hardcoding any type of path
- Adding code to path in your code

```
21 waitForUser("Create/select a folder within which all analysis files will be stored");
22 parent=getDirectory("Select the directory you just created");
23 savefolder=parent+"Raw Images Saved Here" + File.separator;
24 File.makeDirectory(savefolder);
25 processed=parent+"Processed Images Saved Here" + File.separator;
26 File.makeDirectory(processed);
27 moved=parent+"Once Images have Been Processed The Originals are Stored Here" + File.separator;
28 File.makeDirectory(moved);
29 ROIsfolder=parent+"Region Traces Stored Here" + File.separator;
30 File.makeDirectory(ROIsfolder);
31 results=parent+"Results Files are Stored Here" + File.separator;
32 File.makeDirectory(results);
33 microgliamarkers=parent+"Microglia Markers are stored here" + File.separator;
34 File.makeDirectory(microgliamarkers);
35
```

Name	Date modified	Type
Microglia Markers are stored here	10/04/2018 20:10	File folder
Once Images have Been Processed The Originals are Stored Here	10/04/2018 20:10	File folder
Processed Images Saved Here	10/04/2018 20:10	File folder
Raw Images Saved Here	10/04/2018 20:10	File folder
Region Traces Stored Here	10/04/2018 20:10	File folder
Results Files are Stored Here	10/04/2018 20:10	File folder

```
// Create folders necessary to save results
waitForUser("Create/select a folder within which all analysis files will be stored");
parent=getDirectory("Select the directory you just created");

folderNames = newArray( "RawImages",
                        "ProcessedImages",
                        "OrignalImages",
                        "RegionTraces",
                        "ResultFiles",
                        "MicrogliaMarkes");

for (i=0; i<folderNames.length; i++) {
    savefolder=parent+folderNames[i];
    File.makeDirectory(savefolder);
}
```

Content

Part 1:

- Naming Conventions
- Code Layout
- Commenting

Part 2:

- Functionising
- Conceptualisation
- Externalisation

Part 3:

- Debugging
- Testing

brainenergylab.com/sncc

SNCC

SUSSEX NEUROSCIENCE CODING CLUB

This is the temporary website for Sussex Neuroscience Code Club. We aim to upload all the slides and other resources on here as well as any additional information. If you have any questions, email us at:

Schedule

- 30 May – Kira & Dori – Good code part 3: testing and debugging
- 6 June – Devin – SPSS/imaging processing
- 13 June – Ben – Stochastic Programming
- 20 June – Dori – Git/github