

How to generate a kernel

for automated blue whale B call detection

Generate a kernel for spectrogram correlation using at least 30 excellent-quality B calls, each at least 24 hours apart. The kernel is comprised of four segments; this guide will show you how to collect frequency (Hz) values at fixed time intervals (0, 1.5, 3, 4.5, and 10 seconds) and average these values to determine the kernel.

What you need:

- Matlab R2012b
- 'triton1.81-20120218-kernel' – with 'kernel_gen.m' file
- 'loadKernelPicks.m'

June 7, 2013

POC: Sara Kerosky, skerosky@ucsd.edu

1. Update the kernel_gen file in 'triton1.81-20120218-kernel'

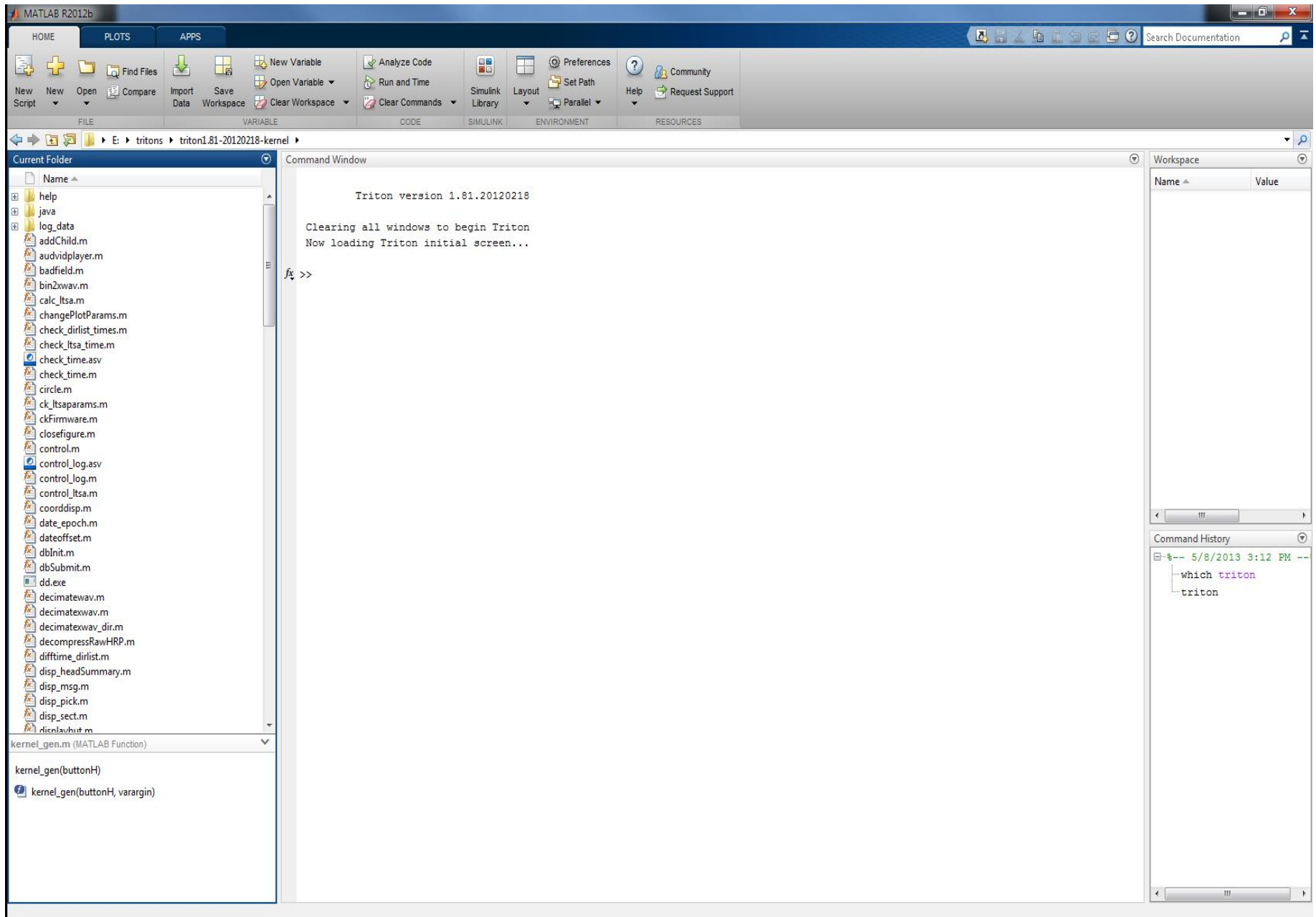
The image shows the MATLAB R2012b environment. The Editor window displays the file `kernel_gen.m` located at `E:\tritons\triton1.81-20120218-kernel\kernel_gen.m`. The file contains the following MATLAB code:

```
30
31 for a = 1:length(box)
32     [C, I] = max(box(:,a)); %where I is the row Index, C is the value
33     Freq(a) = boxFreq(I);
34 end
35
36 %Now pick out time indices of interest....
37 %Find the max of the first 5 points...make that the start.
38
39 firstM = max(Freq(1:10));
40 where = find(Freq == firstM);
41 %Every 5 bins = 1 sec of time. Truncate after 50 pts (ie 10 sec)
42 FreqNew = Freq(5:54);
43
44 Call(1) = FreqNew(1);
45 Call(2) = nanmean(FreqNew(5:10)); %1.5s
46 Call(3) = nanmean(FreqNew(13:18)); %1.5s
47 Call(4) = nanmean(FreqNew(21:26)); %1.5s
48 Call(5) = nanmean(FreqNew(45:50)); %1.5s
49
50 newMatFile = ['SOCAL47M_Bcall.mat'];
51 save(newMatFile, 'Call');
52 disp(['B call characteristics calculated and saved.']);
53
54
55
56
57
58
59
60
61
```

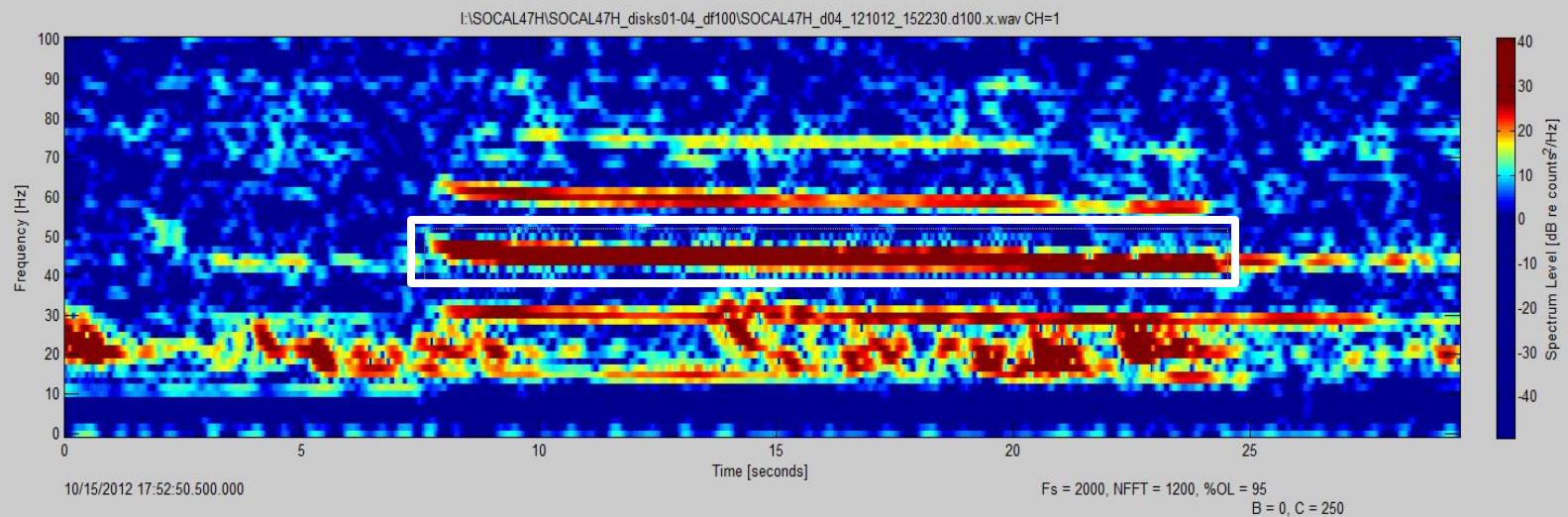
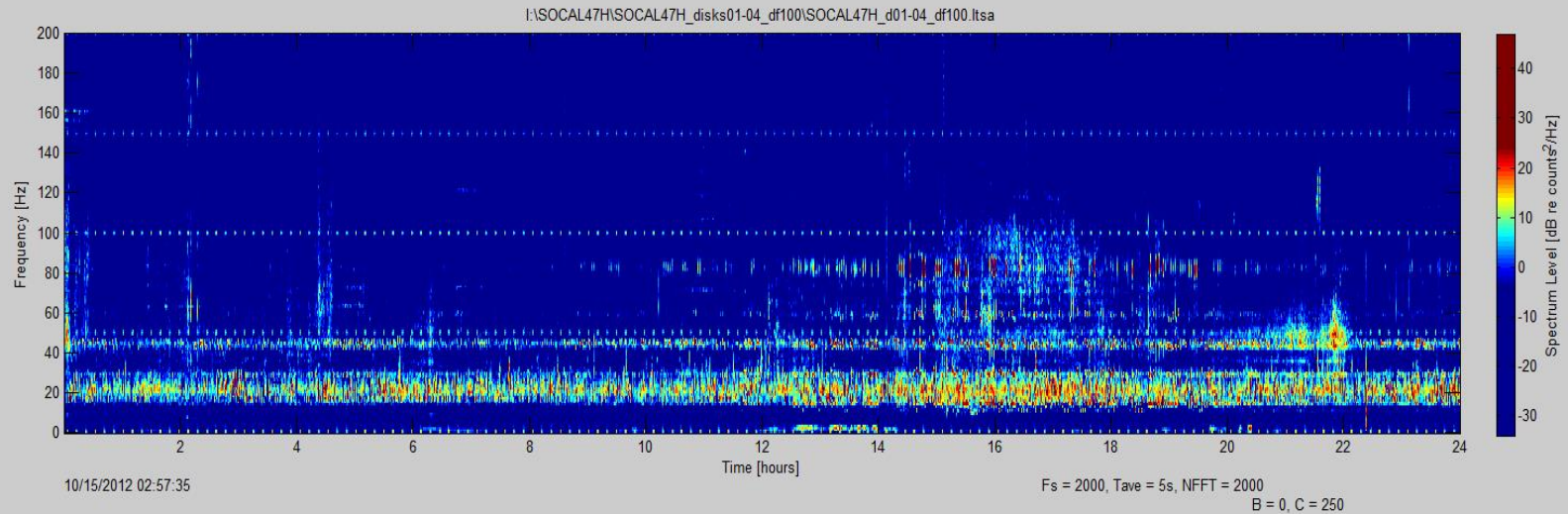
An arrow points to the string `'SOCAL47M'` in line 50, with the text "Update with current deployment information".

The Command Window shows the prompt `>>`. The Workspace window is empty. The Command History window shows the command `--> 5/8/2013 3:12 PM -->`.

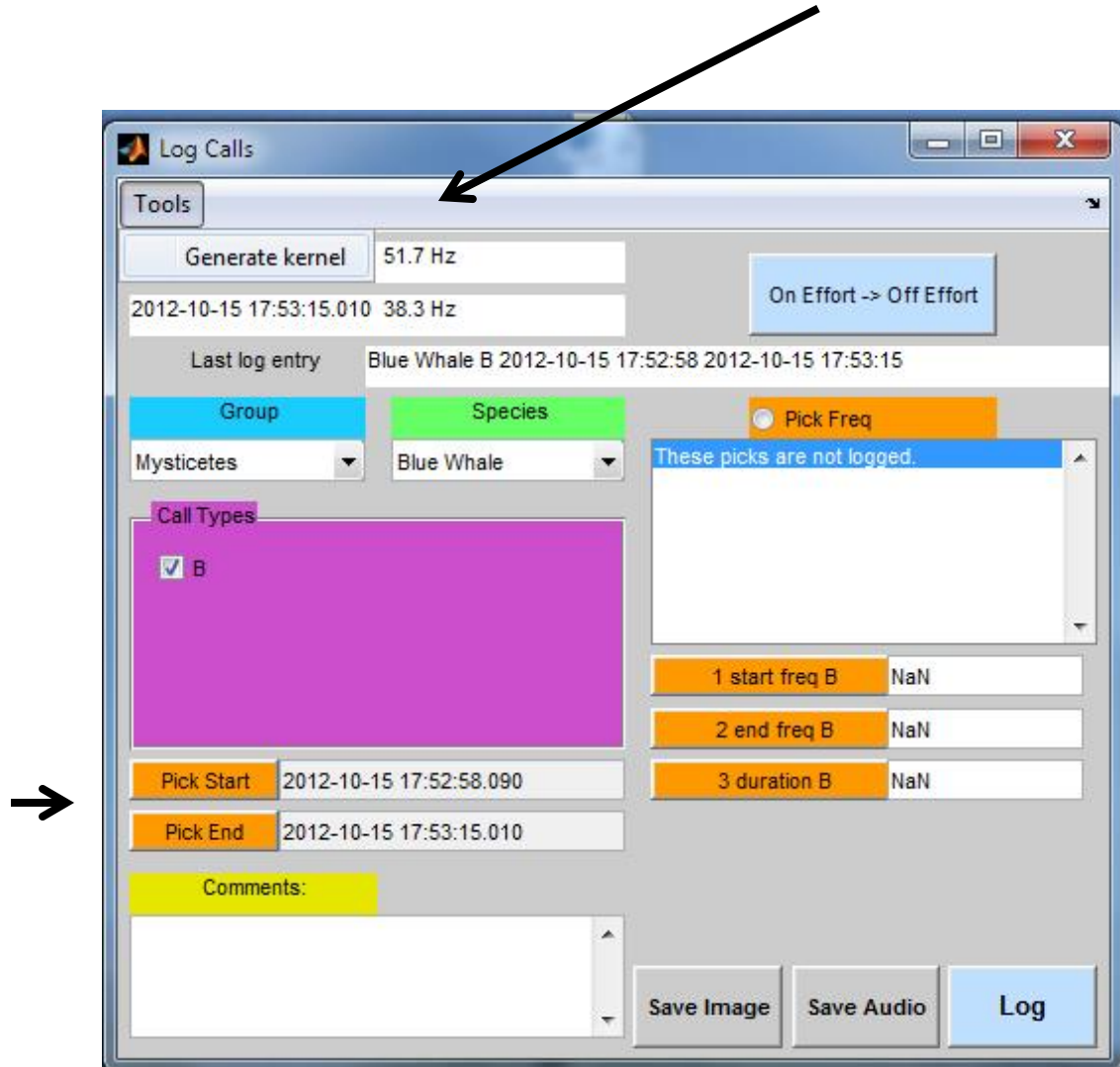
2. Run 'triton1.81-20120218-kernel'



3. Find a nice B call. Using the cursor, draw a box around the call, press 'B' to input timestamps.



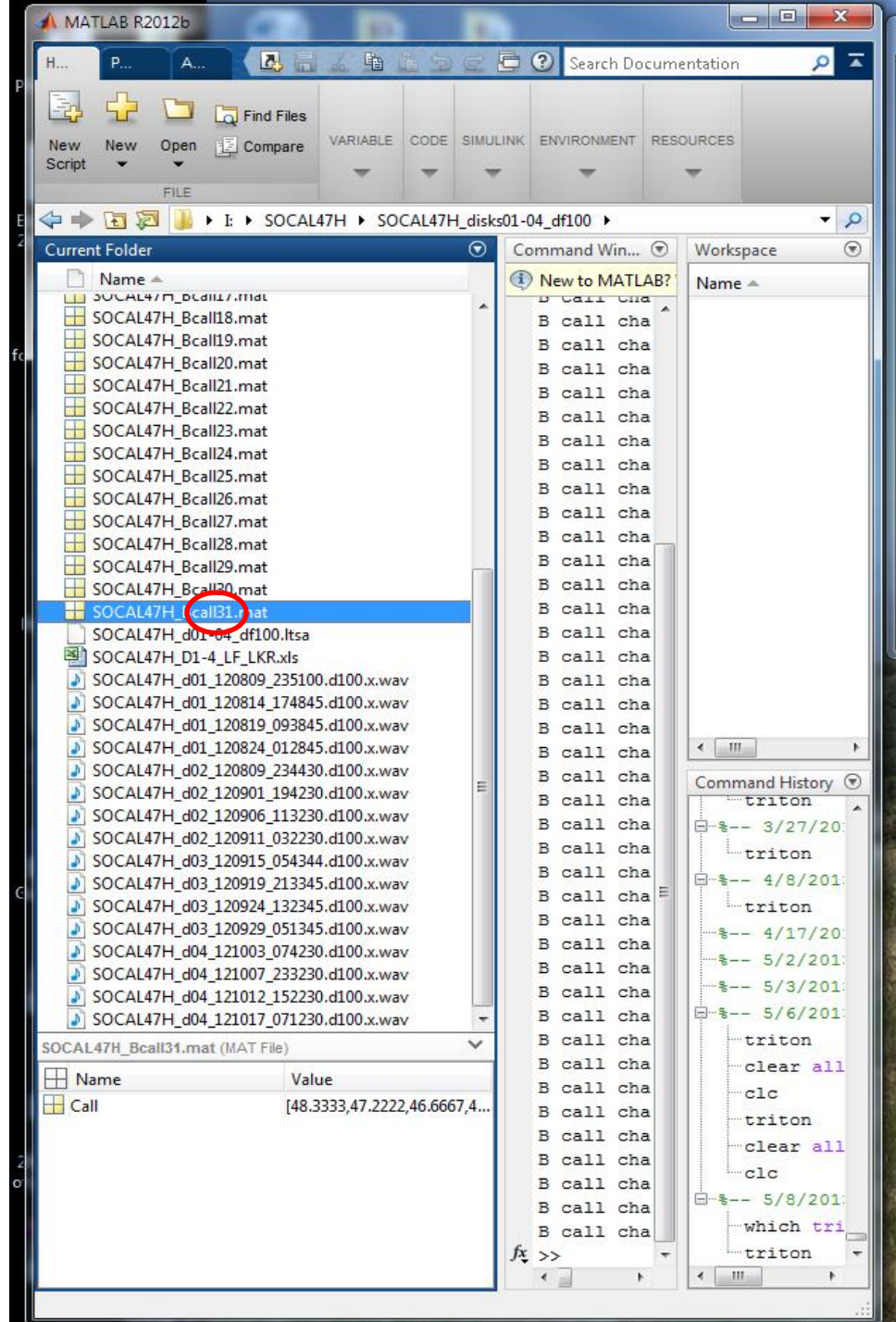
4. Double check the timestamp. Select 'Generate kernel' from the Tools menu.



5. Look at the Current Folder window in Matlab to see the new .mat file created.

IMPORTANT: manually add a number to the .mat filename, otherwise your next call will overwrite these values.

6. Check the values to make sure they make sense.



Log Calls

Tools

Generate kernel 51.7 Hz

2012-10-15 17:53:15.010 38.3 Hz

On Effort -> Off Effort

Last log entry Blue Whale B 2012-10-15 17:52:58 2012-10-15 17:53:15

Group Mysticetes Species Blue Whale

Call Types

☒ B

Pick Start 2012-10-15 17:52:58.090

Pick End 2012-10-15 17:53:15.010

Comments:

1 start freq B NaN

2 end freq B NaN

3 duration B NaN

Save Image Save Audio Log

7. Don't forget to log the call.
Log **AFTER** you check the
values in case you decide to
scrap this call.



8. Once you've finished picking calls, run 'loadKernelPicks'

...it may error out, but that's ok. Just double check that the Calls variable is as big as you expect (number of calls x 5 points). If you are loading calls from multiple deployments, you must...

The screenshot displays the MATLAB IDE interface. On the left, the 'Current Folder' pane shows a list of files, including 'SOCAL46M_kernelpicks_smk-audio', 'SOCAL46M_kernelpicks_smk-image', and various 'SOCAL46M_Bcall' .mat files. The central 'Editor' pane shows the 'loadKernelPicks.m' script. The script includes a directory selection loop and a file loading loop. A red box highlights the line: `filename = 'SOCAL46M_Bcall', num2str(a), '.mat';`. The right 'Workspace' pane shows the 'Calls' variable with a value of '<9x5 double>', which is also highlighted with a red box. The bottom 'Command Window' shows the execution of `loadKernelPicks`, displaying the directory path 'H:\Analysis Detectors\SOCAL 2012 - for ROC' and the number of files '90'. It also shows an error message: 'Error using load Unable to read file SOCAL46M_Bcall10.mat: No such file or directory.' and another error: 'Error in loadKernelPicks (line 31) load(filename)'.

...rename the Calls variable each time to reflect this before you run it again...

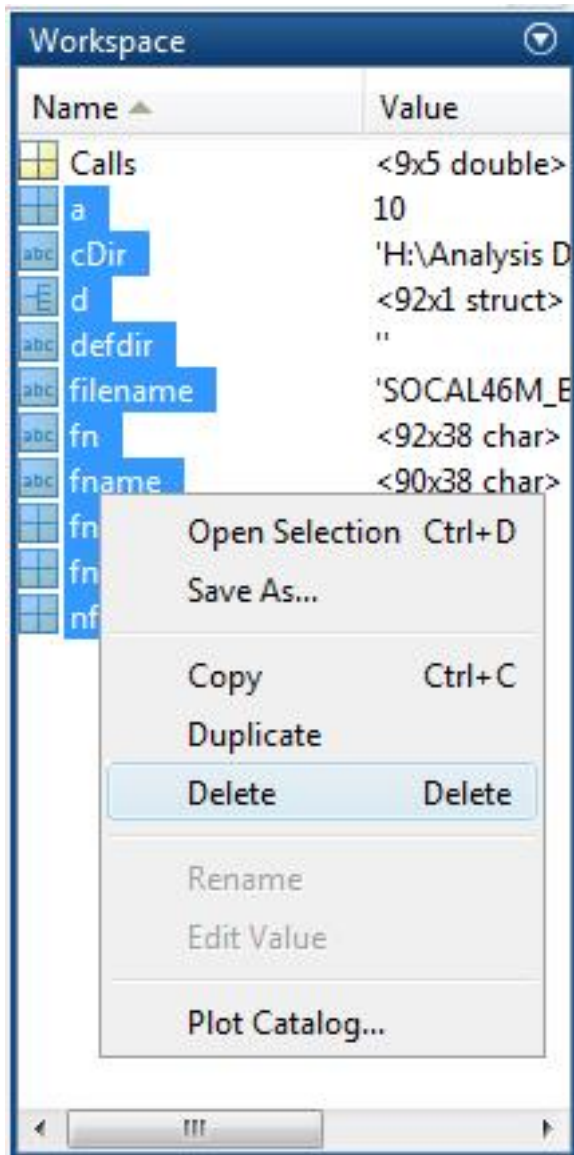
...and update the deployment info so it can find the appropriate .mat files in the directory.

```
1 % Choose directory to find call data
2 defdir = ''; % default directory
3 cDir = uigetdir(defdir, 'Select Directory to Load Call Data');
4 % if the cancel button is pushed, then no file is loaded so exit this script
5 if strcmp(num2str(cDir), '0')
6     disp('Canceled Button Pushed - no directory for call data')
7     return
8 else
9     disp('Load call data from directory : ')
10    disp(cDir)
11 end
12
13 % Display number of files in directory
14 % cDir = 'E:\code\BcallMethod\MATfiles\CallMats'; %Hard code option
15
16 d = dir(cDir); % directory info
17 fn = char(d.name); % file names in directory
18 fnsz = size(fn); % number of files in directory including 2 hidden files
19 fname = fn(3:fnsz(1),:); % get rid of first two hidden files
20 fnamesz = size(fname); % new set of file names
21 nfiles = fnamesz(1); % number of files in directory
22 disp(['Number of files in directory is ', num2str(nfiles)])
23
24 cd(cDir)
25 for a = 1:nfiles
26     filename = 'SOCAL46M_Bcall', num2str(a), '.mat';
27
28     % if exist(filename, 'file') == 2
29
30     load(filename)
31     Calls(a,:) = Call;
32 end
```

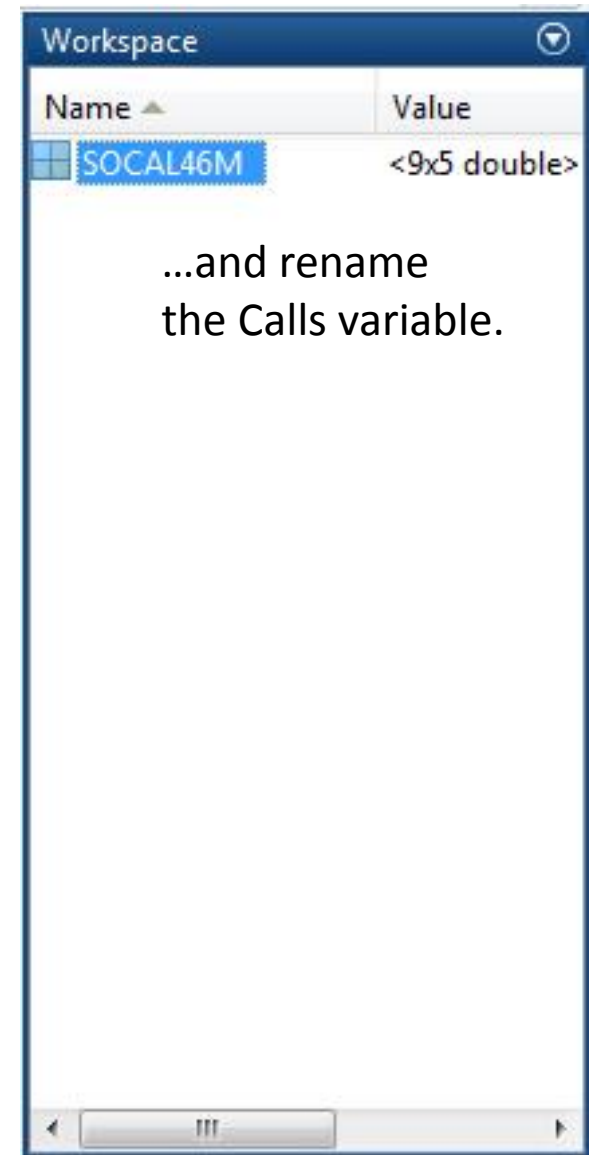
Name	Value
Calls	<9x5 double>
a	10
cDir	'H:\Analysis D
d	<92x1 struct>
defdir	''
filename	'SOCAL46M,B
fn	<92x38 char>
fname	<90x38 char>
fnamesz	[90,38]
fnsz	[92,38]
nfiles	90

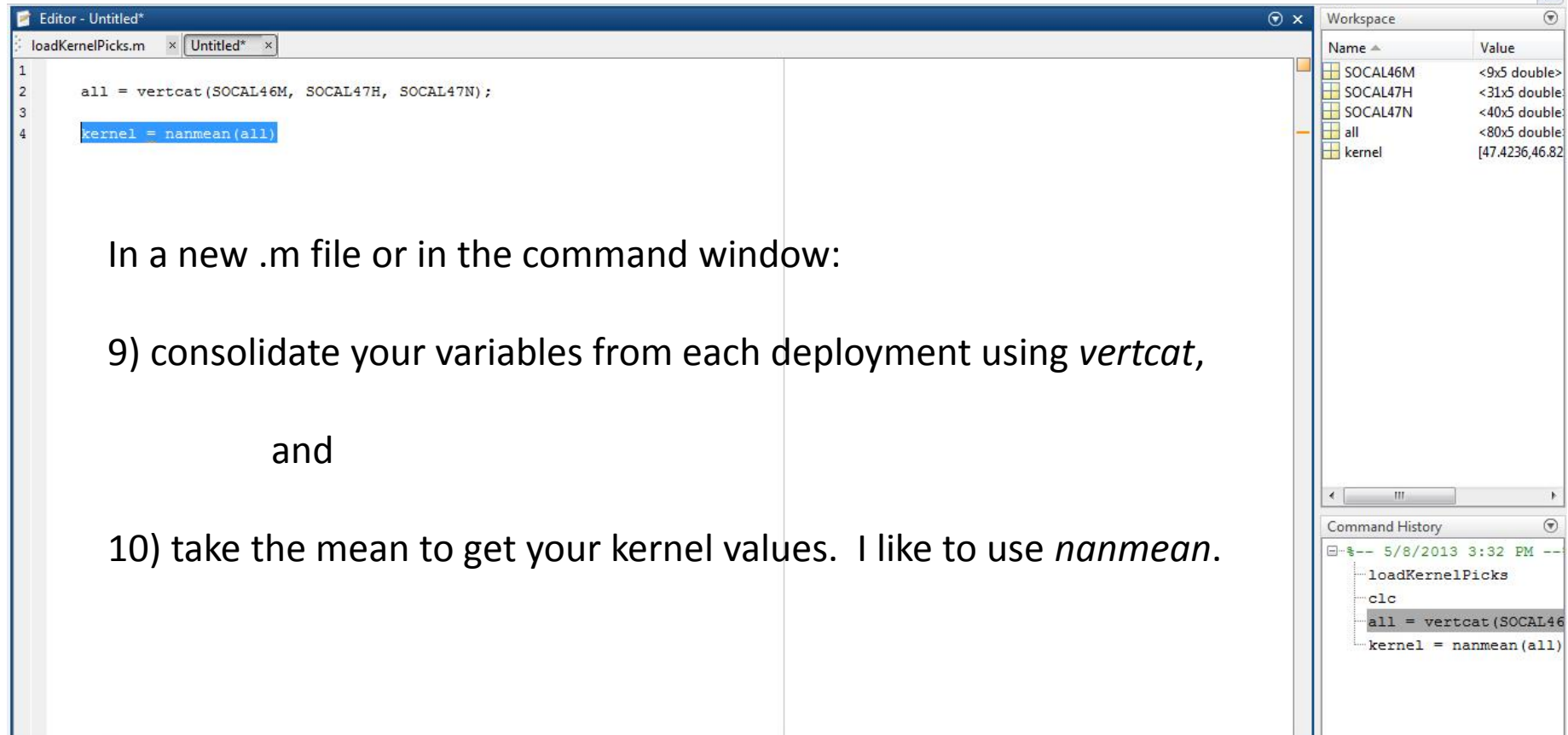
```
>> loadKernelPicks
Load call data from directory :
H:\Analysis Detectors\SOCAL 2012 - for ROC
Number of files in directory is 90
Error using load
Unable to read file SOCAL46M_Bcall10.mat: No such file or directory.

Error in loadKernelPicks (line 31)
    load(filename)
```



Delete the other
variables,





In a new .m file or in the command window:

9) consolidate your variables from each deployment using *vertcat*,

and

10) take the mean to get your kernel values. I like to use *nanmean*.

