

How to run the automated blue whale B call detector

What you need:

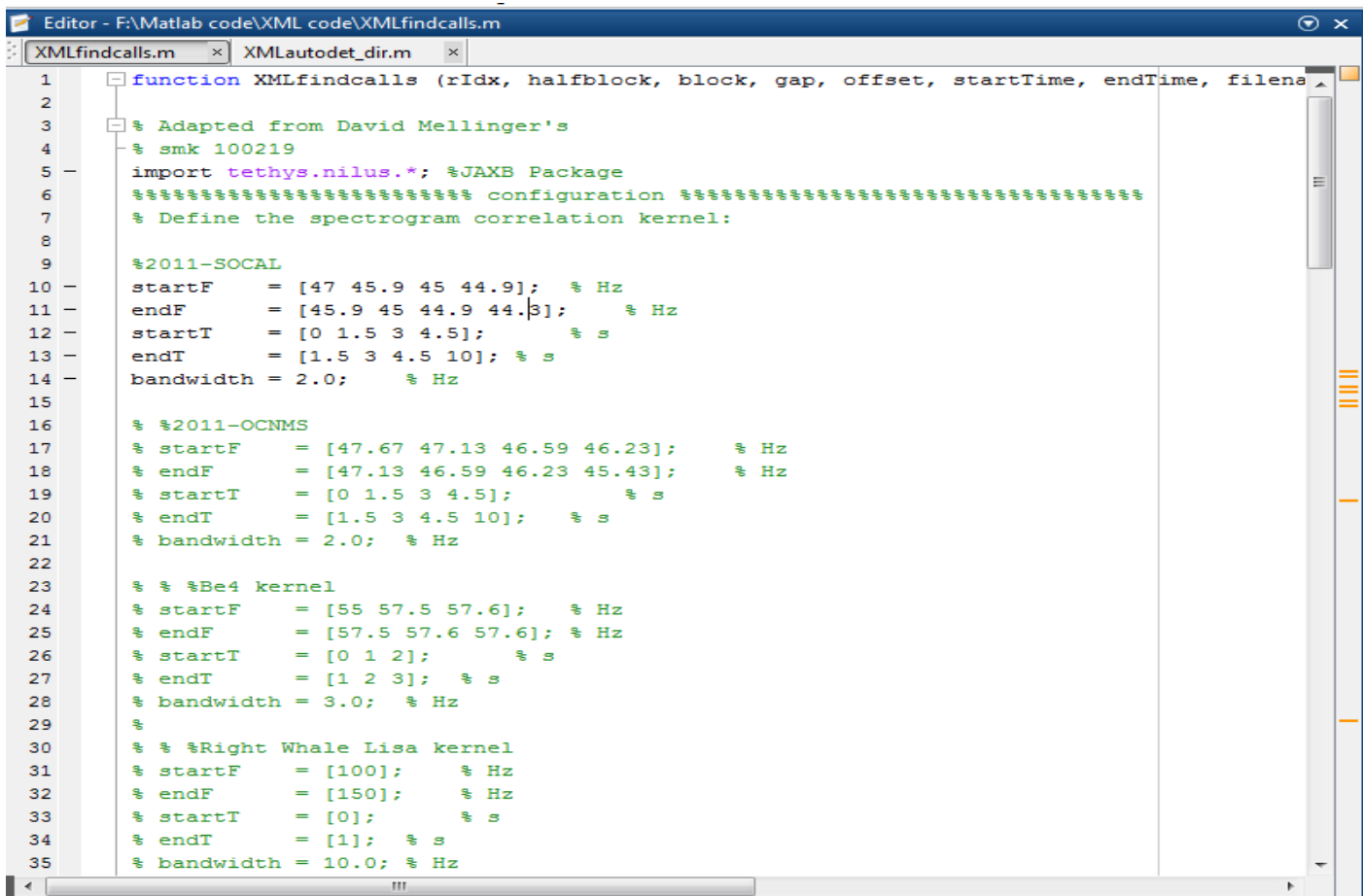
- Matlab R2012b
- triton-2013-11-19
- XML code folder containing
 - XMLfindcalls.m
 - XMLwritecalls.m
 - XMLautodet.m
 - XMLautodet_dir.m
 - nilus.jar

1. Open Matlab and set the path to triton-2013-11-19. You will also need to make sure that Matlab knows where to find the nilus.jar file.

To do this you:

- Set a home directory by typing 'userpath' into the command window of Matlab.
- If this returns blank, set the home directory by typing 'userpath('C:\Users\username\Documents\MATLAB')' where username is replaced by your username. This is a default directory, but any directory can be used.
- Type 'edit javaclasspath.txt' into the command window of Matlab.
- A prompt will ask if you would like to create this file, click yes. If this file already exists, the file will open.
- Add a line to this file that contains the full path to where you have the nilus.jar file (ex: C:\Users\username\Documents\MATLAB\nilus.jar). If you created a new file this will be the only line in the file. If the file already existed, just add an extra line to it.

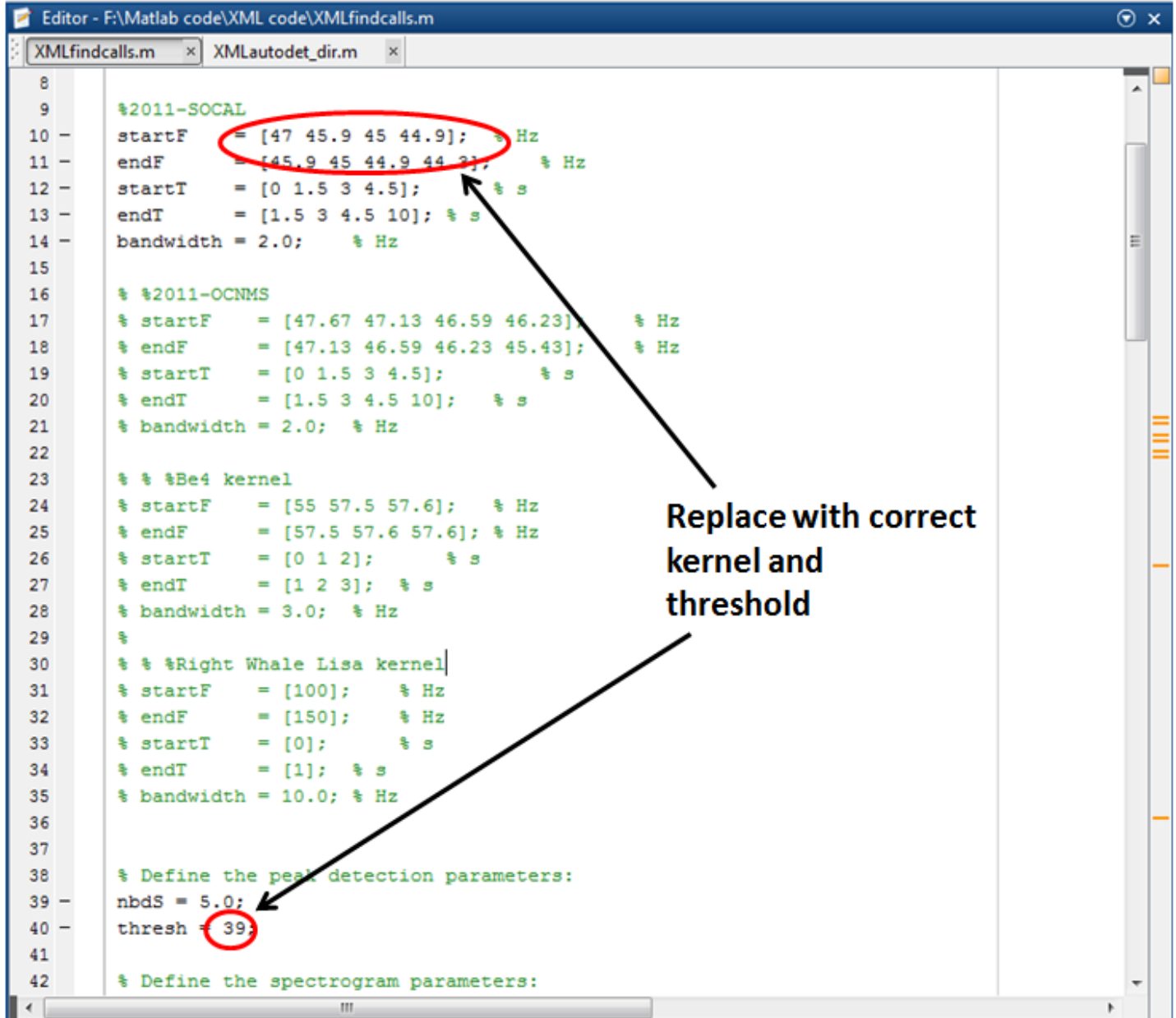
2. Open XMLfindcalls.m and XMLautodet_dir.m in Matlab editor.



The screenshot shows the MATLAB editor window with the file 'XMLfindcalls.m' open. The script is a function that takes several input arguments and defines various spectrogram correlation kernels. The code is as follows:

```
1 function XMLfindcalls (rIdx, halfblock, block, gap, offset, startTime, endTime, fileName)
2
3 % Adapted from David Mellinger's
4 % smk 100219
5 import tethys.nilus.*; %JAXB Package
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% configuration %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 % Define the spectrogram correlation kernel:
8
9 %2011-SOCAL
10 startF = [47 45.9 45 44.9]; % Hz
11 endF = [45.9 45 44.9 44.3]; % Hz
12 startT = [0 1.5 3 4.5]; % s
13 endT = [1.5 3 4.5 10]; % s
14 bandwidth = 2.0; % Hz
15
16 % %2011-OCNMS
17 % startF = [47.67 47.13 46.59 46.23]; % Hz
18 % endF = [47.13 46.59 46.23 45.43]; % Hz
19 % startT = [0 1.5 3 4.5]; % s
20 % endT = [1.5 3 4.5 10]; % s
21 % bandwidth = 2.0; % Hz
22
23 % %Be4 kernel
24 % startF = [55 57.5 57.6]; % Hz
25 % endF = [57.5 57.6 57.6]; % Hz
26 % startT = [0 1 2]; % s
27 % endT = [1 2 3]; % s
28 % bandwidth = 3.0; % Hz
29 %
30 % %Right Whale Lisa kernel
31 % startF = [100]; % Hz
32 % endF = [150]; % Hz
33 % startT = [0]; % s
34 % endT = [1]; % s
35 % bandwidth = 10.0; % Hz
```

3. Click on XMLfindcalls.m and enter the correct kernel information (the same way you did when trying to determine the optimal threshold). Then scroll down and enter the threshold.



```
8
9 %2011-SOCAL
10 startF = [47 45.9 45 44.9]; % Hz
11 endF = [45.9 45 44.9 44.2]; % Hz
12 startT = [0 1.5 3 4.5]; % s
13 endT = [1.5 3 4.5 10]; % s
14 bandwidth = 2.0; % Hz
15
16 % %2011-OCNMS
17 % startF = [47.67 47.13 46.59 46.23]; % Hz
18 % endF = [47.13 46.59 46.23 45.43]; % Hz
19 % startT = [0 1.5 3 4.5]; % s
20 % endT = [1.5 3 4.5 10]; % s
21 % bandwidth = 2.0; % Hz
22
23 % % %Be4 kernel
24 % startF = [55 57.5 57.6]; % Hz
25 % endF = [57.5 57.6 57.6]; % Hz
26 % startT = [0 1 2]; % s
27 % endT = [1 2 3]; % s
28 % bandwidth = 3.0; % Hz
29 %
30 % % %Right Whale Lisa kernel
31 % startF = [100]; % Hz
32 % endF = [150]; % Hz
33 % startT = [0]; % s
34 % endT = [1]; % s
35 % bandwidth = 10.0; % Hz
36
37
38 % Define the peak detection parameters:
39 nbdS = 5.0;
40 thresh = 39;
41
42 % Define the spectrogram parameters:
```

4. **SAVE** XMLfindcalls.m.

- Click on XMLautodet_dir.m. Enter the name you would like the xls output file to be called (include the deployment info, the kernel name and the threshold) and below that enter the location where you would like the xls file to be placed. Then scroll down and enter the same information for the xml version of the file (this is set up so that you enter the location followed by a \ and then the name of the file).

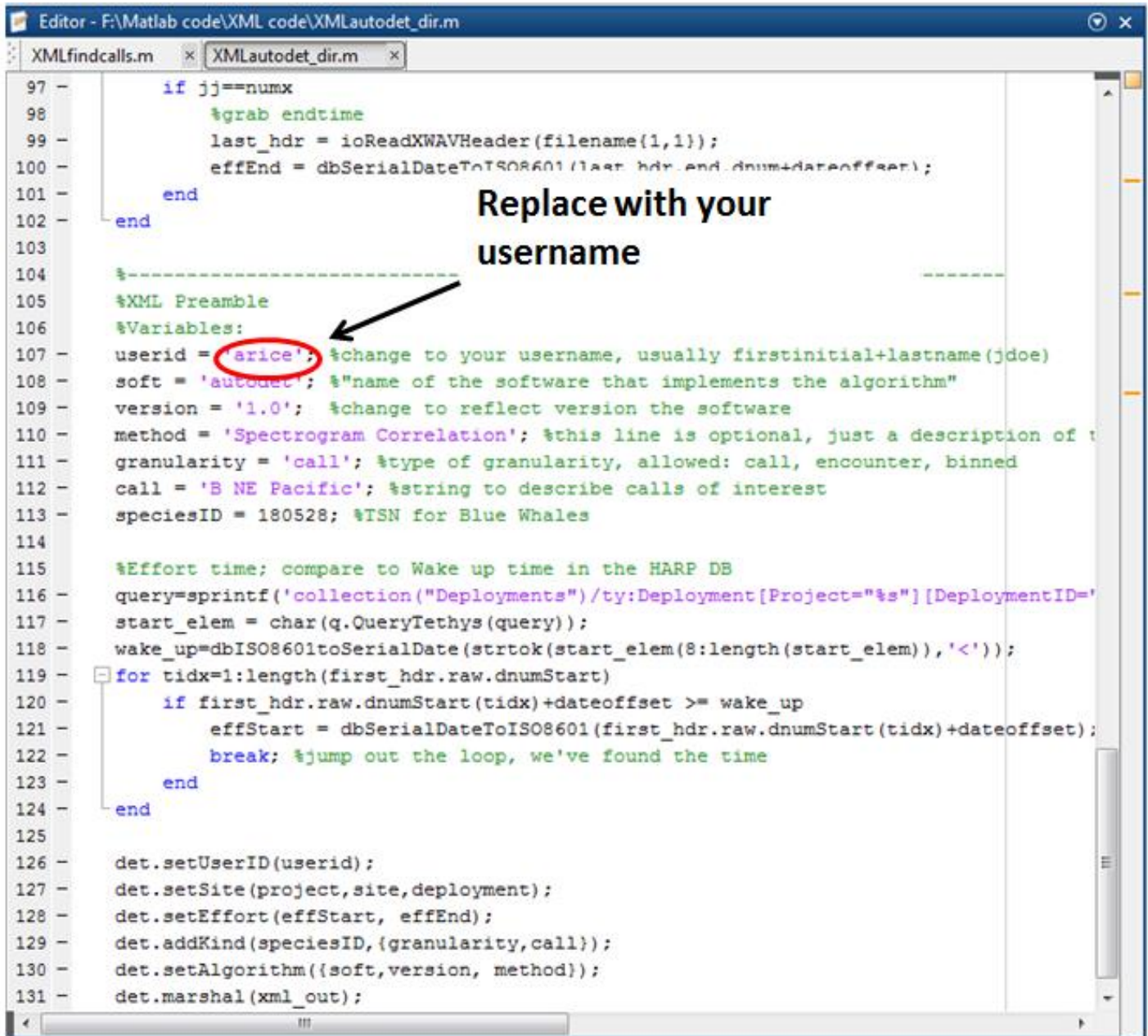
```
37
38 %-----
39 % Choose output file
40 %boxTitle1 = ['Select Output Text File'];
41 %filterSpec1 = '*.txt';
42 %defaultName = 'E:';
43 % [outfile, outpath] = uigetfile(filterSpec1, boxTitle1, defaultName);
44 outfile = 'SOCAL45H_ker45M_thresh39.xls';
45 outpath = 'F:\Detections\Site H\Raw detections';
46 outflen = length(outfile);
47 outplen = length(outpath);
48
49 %-----
50 % Choose XML output file
51 % boxTitle2 = ['Select Output XML File'];
52 % filterSpec2 = {'*.xml'; '*.txt'};
53 % defaultName2 = 'E:';
54 % [xmlfile, xmlpath] = uigetfile(filterSpec2, boxTitle2, defaultName2);
55 % xml_out = strcat(xmlpath, xmlfile);
56
57 %hardcoded location of output XML. if not using prompt
58 xml_out = 'F:\Detections\Site H\Raw detections\SOCAL45H_ker45M_thresh39.xml';
59
60 %-----
61 % Loop the following codes through chosen directory
62 detALL=[];
63 det=Detections(); %create XML object input
64 proc_dir=true; %tell autodet it's working through a directory
65 for jj = 1:numx
66     directory.inpath = strcat(idir, '\');
67     directory.infiledet = xwaves(jj,:); % get file names sequentially
68     filename = strcat(directory.inpath, directory.infiledet);
69     disp(['Looking for calls in ' filename{1,1}])
70     % [fid,message] = fopen([xwaves.inpath, xwaves.infiledet], 'r'); %reading file
71     if jj==1 %first run
```

Replace with xls file name and location

Replace with xml file location

Replace with xml file name

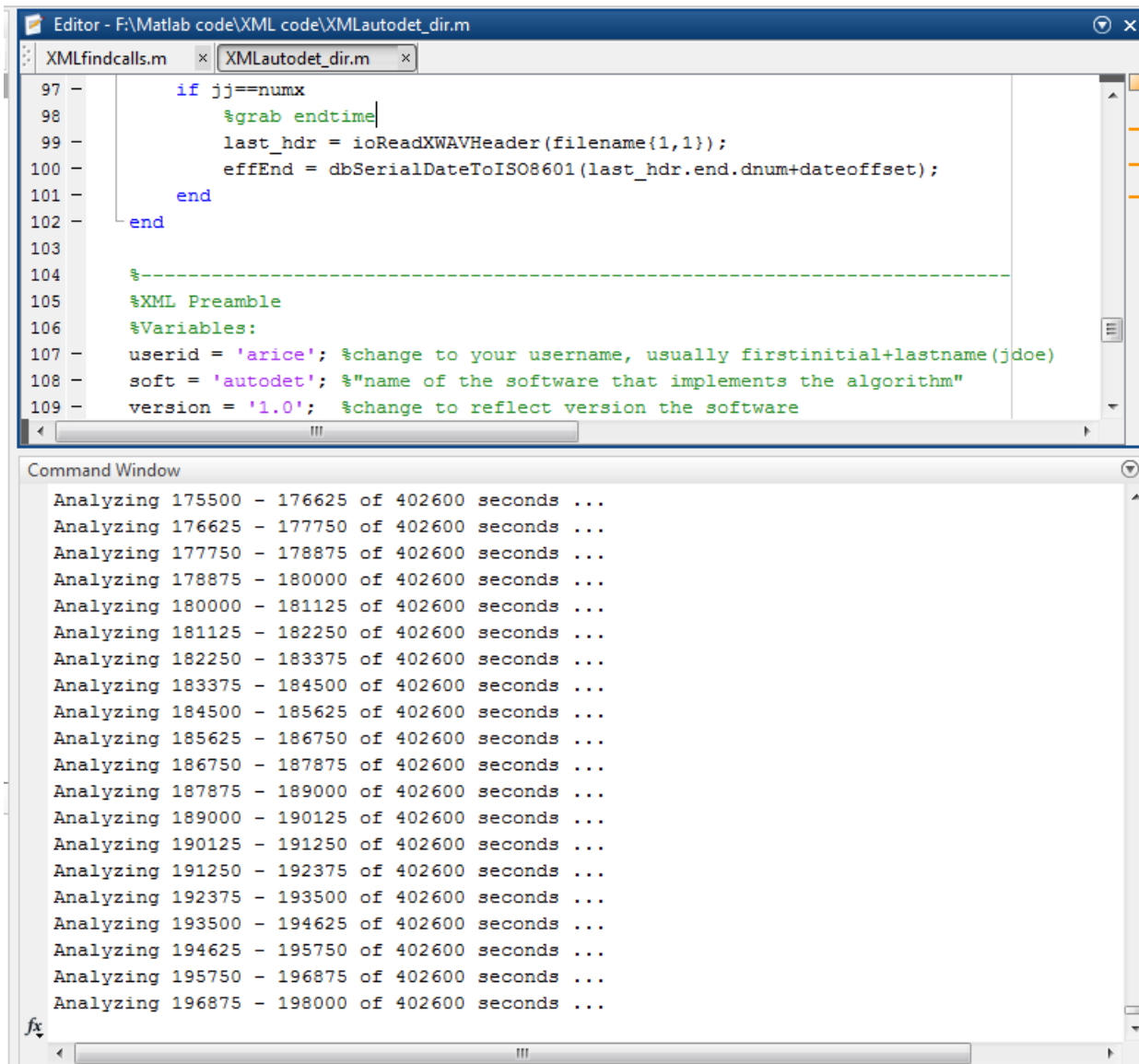
- Then, scroll down further and change the current username to your username. You should only have to change the username once (when you save/run XMLautodet_dir.m your username will be saved along with it).



```
Editor - F:\Matlab code\XML code\XMLautodet_dir.m
XMLfindcalls.m x XMLautodet_dir.m x

97 -         if jj==numx
98             %grab endtime
99             last_hdr = ioReadXWAVHeader(filename{1,1});
100             effEnd = dbSerialDateToISO8601(last_hdr.end.dnum+dateoffset);
101         end
102     end
103
104     %-----
105     %XML Preamble
106     %Variables:
107     userid = 'arice'; %change to your username, usually firstinitial+lastname(jdoe)
108     soft = 'autodet'; %"name of the software that implements the algorithm"
109     version = '1.0'; %change to reflect version the software
110     method = 'Spectrogram Correlation'; %this line is optional, just a description of t
111     granularity = 'call'; %type of granularity, allowed: call, encounter, binned
112     call = 'B NE Pacific'; %string to describe calls of interest
113     speciesID = 180528; %TSN for Blue Whales
114
115     %Effort time; compare to Wake up time in the HARP DB
116     query=sprintf('collection("Deployments")/ty:Deployment[Project="%s"] [DeploymentID='
117     start_elem = char(q.QueryTethys(query));
118     wake_up=dbISO8601toSerialDate(strtok(start_elem(8:length(start_elem)), '<'));
119     for tid=1:length(first_hdr.raw.dnumStart)
120         if first_hdr.raw.dnumStart(tid)+dateoffset >= wake_up
121             effStart = dbSerialDateToISO8601(first_hdr.raw.dnumStart(tid)+dateoffset);
122             break; %jump out the loop, we've found the time
123         end
124     end
125
126     det.setUserID(userid);
127     det.setSite(project,site,deployment);
128     det.setEffort(effStart, effEnd);
129     det.addKind(speciesID,{granularity,call});
130     det.setAlgorithm({soft,version, method});
131     det.marshall(xml_out);
```

7. Run XMLautodet_dir.m. This will take a while.



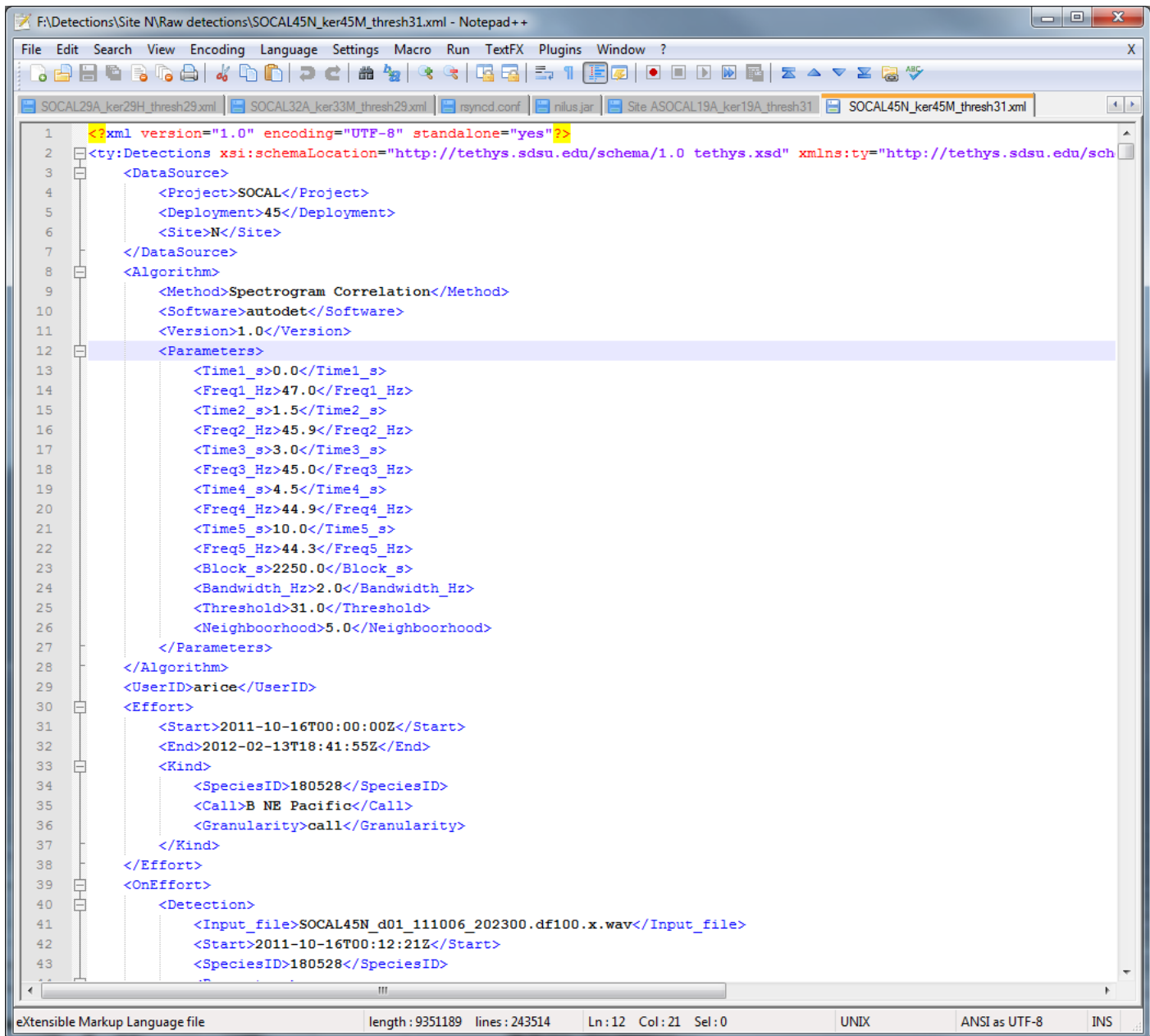
The screenshot shows the MATLAB Editor window with the file `XMLautodet_dir.m` open. The code in the editor includes a loop structure with an `if` statement and a `%grab endtime` comment. Below the loop, there is a section for XML metadata including a preamble, variables, and user-defined fields like `userid`, `soft`, and `version`.

```
97 -         if jj==numx
98 -             %grab endtime
99 -             last_hdr = ioReadXWAVHeader(filename{1,1});
100 -             effEnd = dbSerialDateToISO8601(last_hdr.end.dnum+dateoffset);
101 -         end
102 -     end
103
104 -     %-----
105 -     %XML Preamble
106 -     %Variables:
107 -     userid = 'arice'; %change to your username, usually firstinitial+lastname(jdoe)
108 -     soft = 'autodet'; %"name of the software that implements the algorithm"
109 -     version = '1.0'; %change to reflect version the software
```

The Command Window displays the progress of the analysis, showing a series of time intervals being processed. The output is as follows:

```
Analyzing 175500 - 176625 of 402600 seconds ...
Analyzing 176625 - 177750 of 402600 seconds ...
Analyzing 177750 - 178875 of 402600 seconds ...
Analyzing 178875 - 180000 of 402600 seconds ...
Analyzing 180000 - 181125 of 402600 seconds ...
Analyzing 181125 - 182250 of 402600 seconds ...
Analyzing 182250 - 183375 of 402600 seconds ...
Analyzing 183375 - 184500 of 402600 seconds ...
Analyzing 184500 - 185625 of 402600 seconds ...
Analyzing 185625 - 186750 of 402600 seconds ...
Analyzing 186750 - 187875 of 402600 seconds ...
Analyzing 187875 - 189000 of 402600 seconds ...
Analyzing 189000 - 190125 of 402600 seconds ...
Analyzing 190125 - 191250 of 402600 seconds ...
Analyzing 191250 - 192375 of 402600 seconds ...
Analyzing 192375 - 193500 of 402600 seconds ...
Analyzing 193500 - 194625 of 402600 seconds ...
Analyzing 194625 - 195750 of 402600 seconds ...
Analyzing 195750 - 196875 of 402600 seconds ...
Analyzing 196875 - 198000 of 402600 seconds ...
```

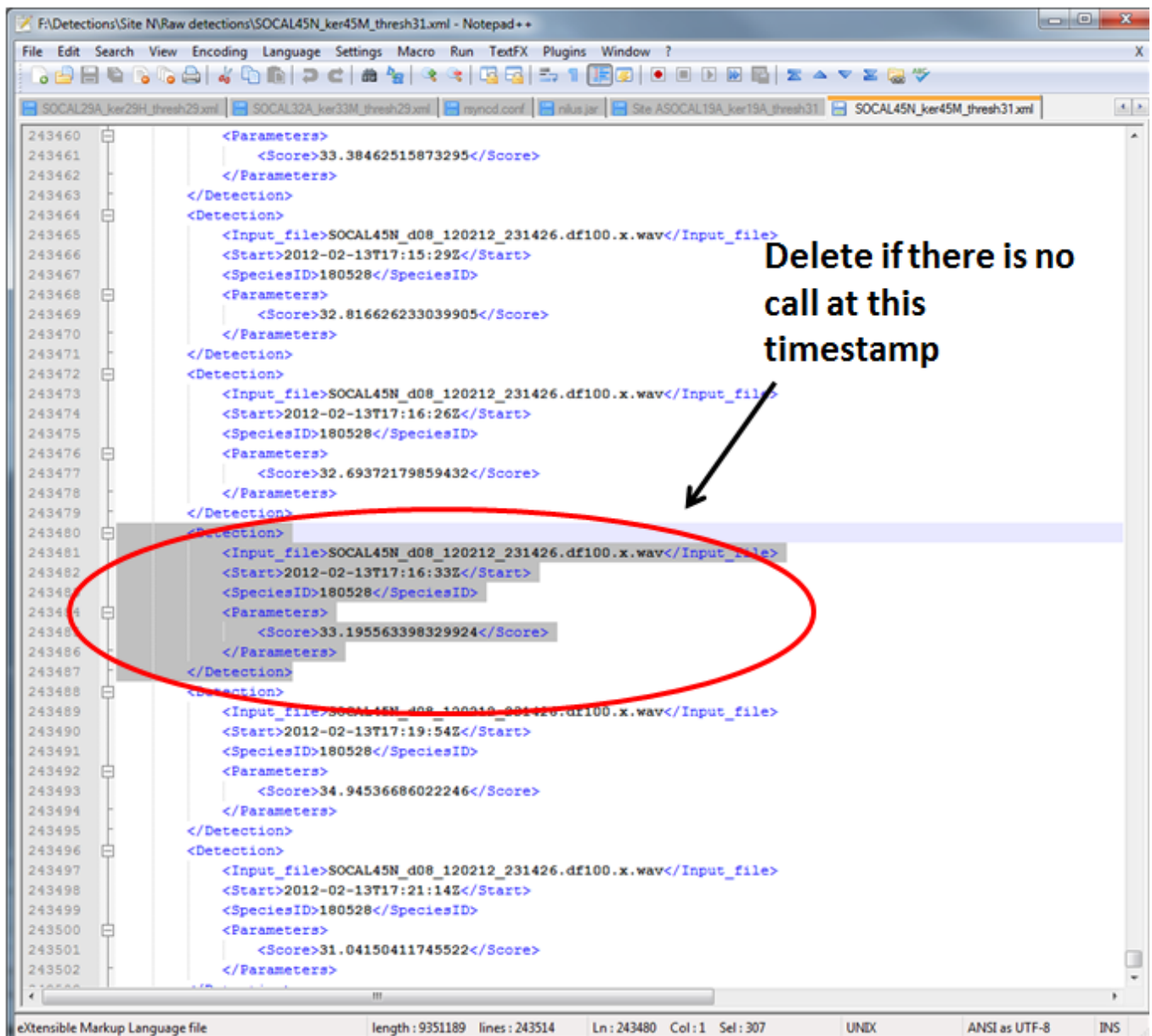
8. After the detector has finished running, right click on the newly created XML file and click 'edit in Notepad++'. (If your computer does not have Notepad you can download it here: <http://notepad-plus-plus.org/download/v6.5.1.html>.) This will open the XML version of the automatic detections.



```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <ty:Detections xsi:schemaLocation="http://tethys.sdsu.edu/schema/1.0 tethys.xsd" xmlns:ty="http://tethys.sdsu.edu/sch
3  <DataSource>
4      <Project>SOCAL</Project>
5      <Deployment>45</Deployment>
6      <Site>N</Site>
7  </DataSource>
8  <Algorithm>
9      <Method>Spectrogram Correlation</Method>
10     <Software>autodet</Software>
11     <Version>1.0</Version>
12     <Parameters>
13         <Time1_s>0.0</Time1_s>
14         <Freq1_Hz>47.0</Freq1_Hz>
15         <Time2_s>1.5</Time2_s>
16         <Freq2_Hz>45.9</Freq2_Hz>
17         <Time3_s>3.0</Time3_s>
18         <Freq3_Hz>45.0</Freq3_Hz>
19         <Time4_s>4.5</Time4_s>
20         <Freq4_Hz>44.9</Freq4_Hz>
21         <Time5_s>10.0</Time5_s>
22         <Freq5_Hz>44.3</Freq5_Hz>
23         <Block_s>2250.0</Block_s>
24         <Bandwidth_Hz>2.0</Bandwidth_Hz>
25         <Threshold>31.0</Threshold>
26         <Neighborhood>5.0</Neighborhood>
27     </Parameters>
28 </Algorithm>
29 <UserID>arice</UserID>
30 <Effort>
31     <Start>2011-10-16T00:00:00Z</Start>
32     <End>2012-02-13T18:41:55Z</End>
33     <Kind>
34         <SpeciesID>180528</SpeciesID>
35         <Call>B NE Pacific</Call>
36         <Granularity>call</Granularity>
37     </Kind>
38 </Effort>
39 <OnEffort>
40     <Detection>
41         <Input_file>SOCAL45N_d01_111006_202300.df100.x.wav</Input_file>
42         <Start>2011-10-16T00:12:21Z</Start>
43         <SpeciesID>180528</SpeciesID>
```

eXtensible Markup Language file length: 9351189 lines: 243514 Ln: 12 Col: 21 Sel: 0 UNIX ANSI as UTF-8 INS

9. Detections that occur from Feb – May will need to be manually checked for false detections. So, if your xml file has detections during those months you will need to open triton and check if there actually were calls for those detections. If you find that there is no call for a detection then you need to delete that entry in the xml file.



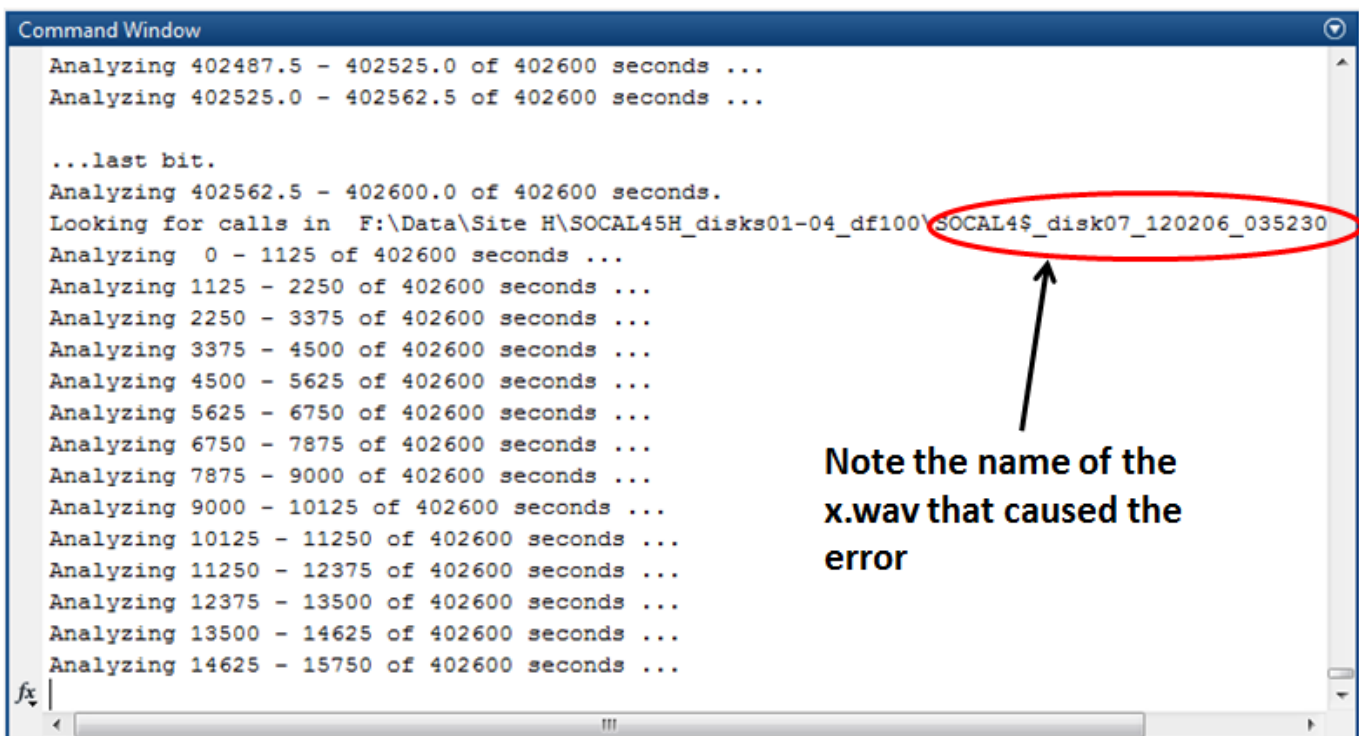
10. Whether or not you checked/deleted detections, **SAVE** the edited xml into a new folder (you don't want to overwrite the unedited detections). For each automatic detection you want to have a raw detection file and a checked detection file (representing what the detector gave you and what you're actually going to use for call count analysis, respectively).

11. Once you have an xml file that has been checked for false calls, if necessary, it can be submitted to the database.

Common error message:

If the detector stops running and you get an error message that starts “corrupt xwav file...” this means that there was a problem with one of the x.wav files you were trying to run. So, you will need to run the detector only on the x.wavs that come after the x.wav that had the error. In order to do this you need to:

1. Make a note of which file it was that had the error.



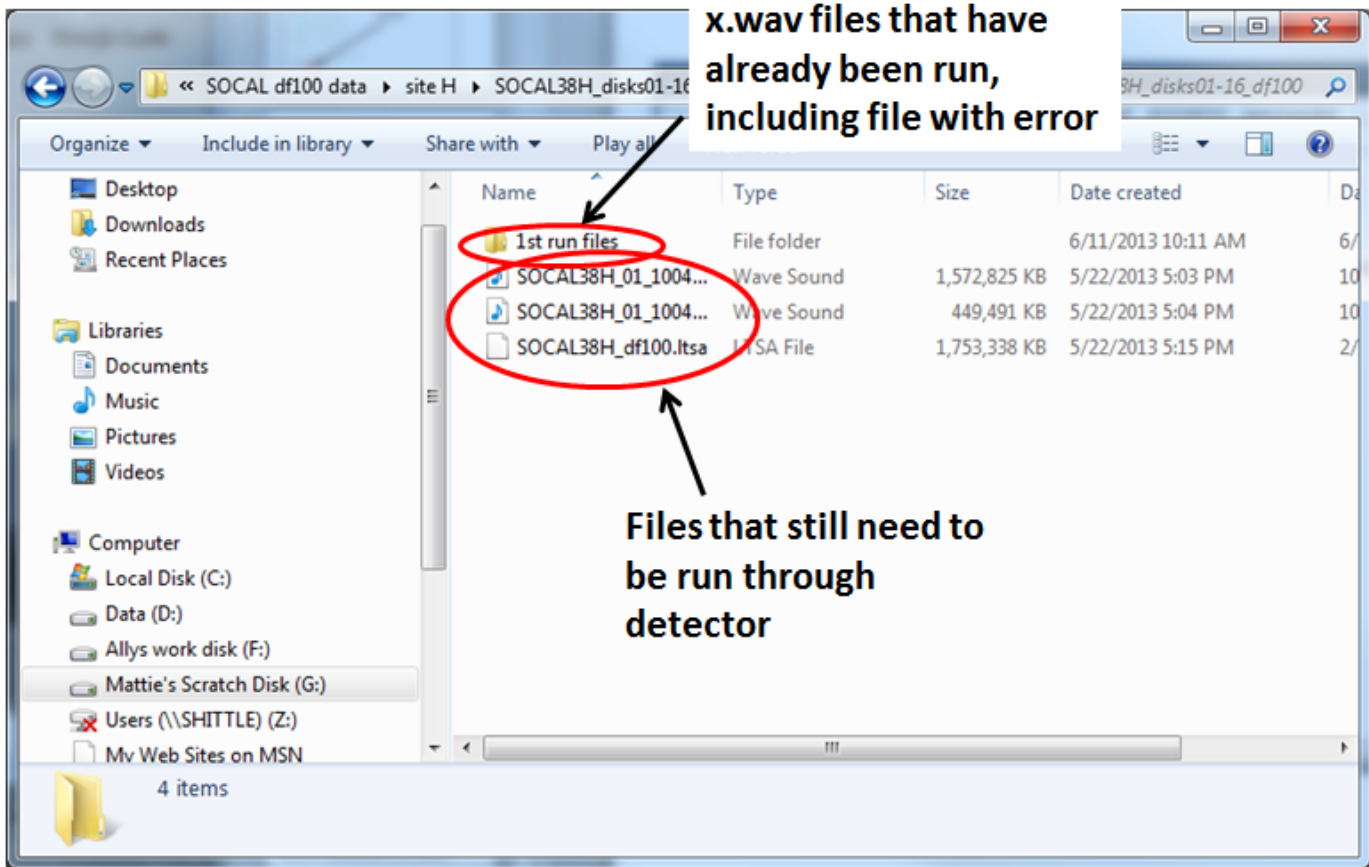
```
Command Window
Analyzing 402487.5 - 402525.0 of 402600 seconds ...
Analyzing 402525.0 - 402562.5 of 402600 seconds ...

...last bit.
Analyzing 402562.5 - 402600.0 of 402600 seconds.
Looking for calls in F:\Data\Site H\SOCAL45H_disks01-04_df100\SOCAL4$_disk07_120206_035230
Analyzing 0 - 1125 of 402600 seconds ...
Analyzing 1125 - 2250 of 402600 seconds ...
Analyzing 2250 - 3375 of 402600 seconds ...
Analyzing 3375 - 4500 of 402600 seconds ...
Analyzing 4500 - 5625 of 402600 seconds ...
Analyzing 5625 - 6750 of 402600 seconds ...
Analyzing 6750 - 7875 of 402600 seconds ...
Analyzing 7875 - 9000 of 402600 seconds ...
Analyzing 9000 - 10125 of 402600 seconds ...
Analyzing 10125 - 11250 of 402600 seconds ...
Analyzing 11250 - 12375 of 402600 seconds ...
Analyzing 12375 - 13500 of 402600 seconds ...
Analyzing 13500 - 14625 of 402600 seconds ...
Analyzing 14625 - 15750 of 402600 seconds ...
```

Note the name of the x.wav that caused the error

2. Close Matlab
3. Go into the directory of x.wav files and create a new folder.

4. Place the x.wav containing the error and all of the x.wavs before it into the folder you just created.



5. Start Matlab.
6. In XMLautodet_dir.m you will need to change the name of the xls and xml file you are saving your detection as to something different, so that your previous file isn't written over (ex: SOCAL45H_ker45M_thresh39-part2).
7. Run XMLautodet_dir.m again.