How to run the automated blue whale B call detector

What you need:

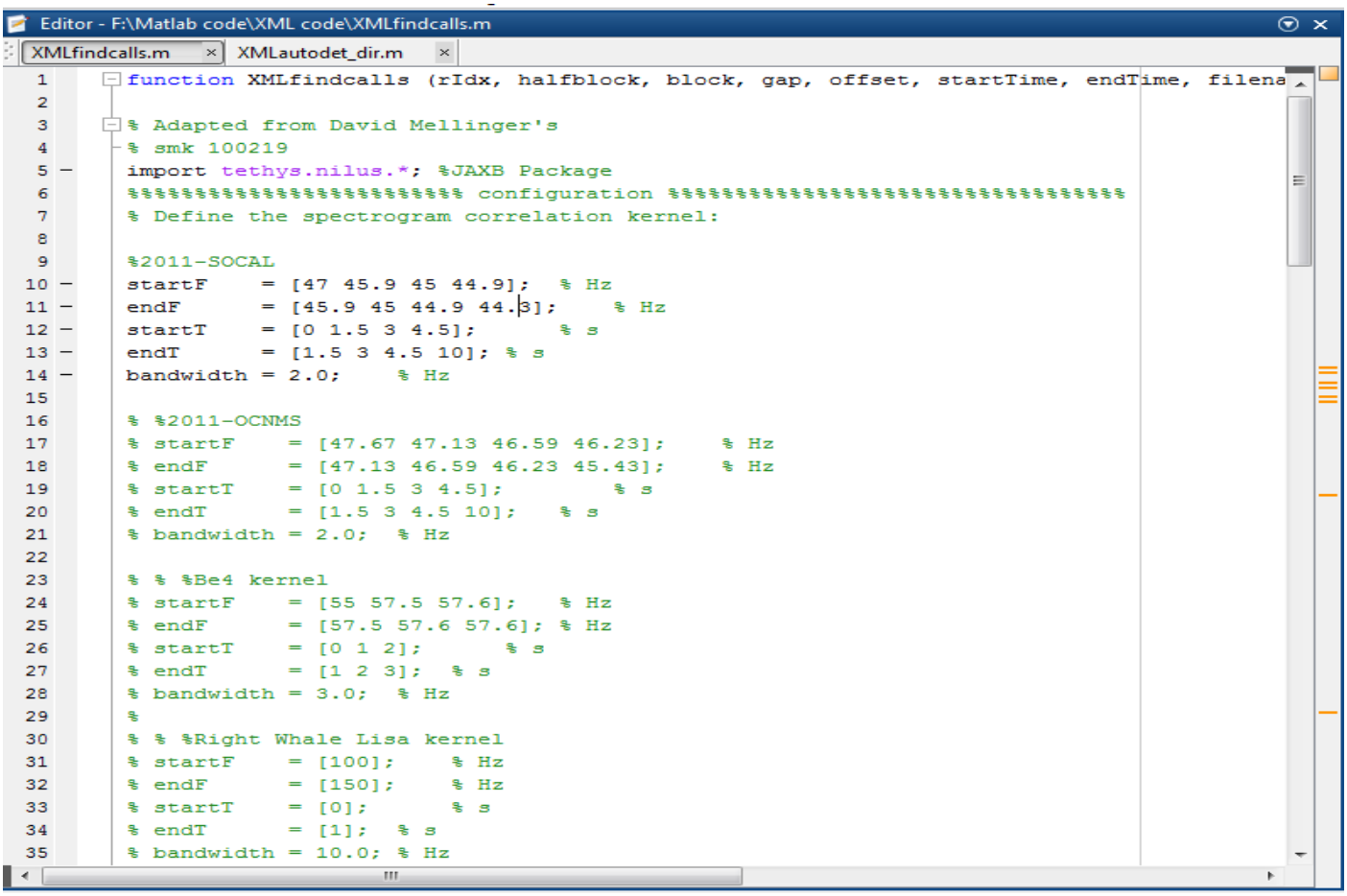
- Matlab R2012b
- triton-2013-11-19
- XML code folder containing
    - XMLfindcalls.m
    - XMLwritecalls.m
    - XMLautodet.m
    - XMLautodet_dir.m
    - nilus.jar

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.

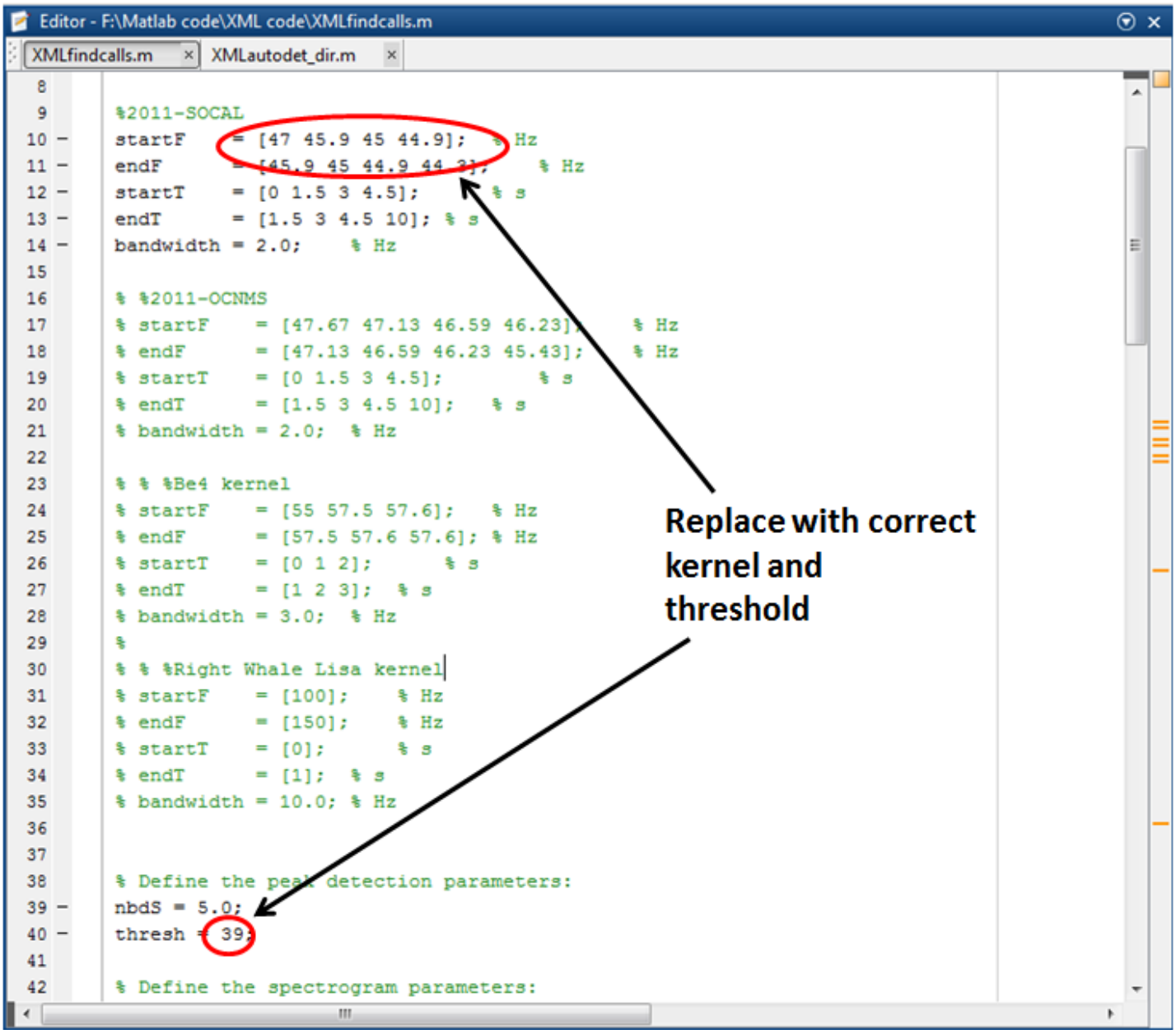
   To do this you:

    - 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.

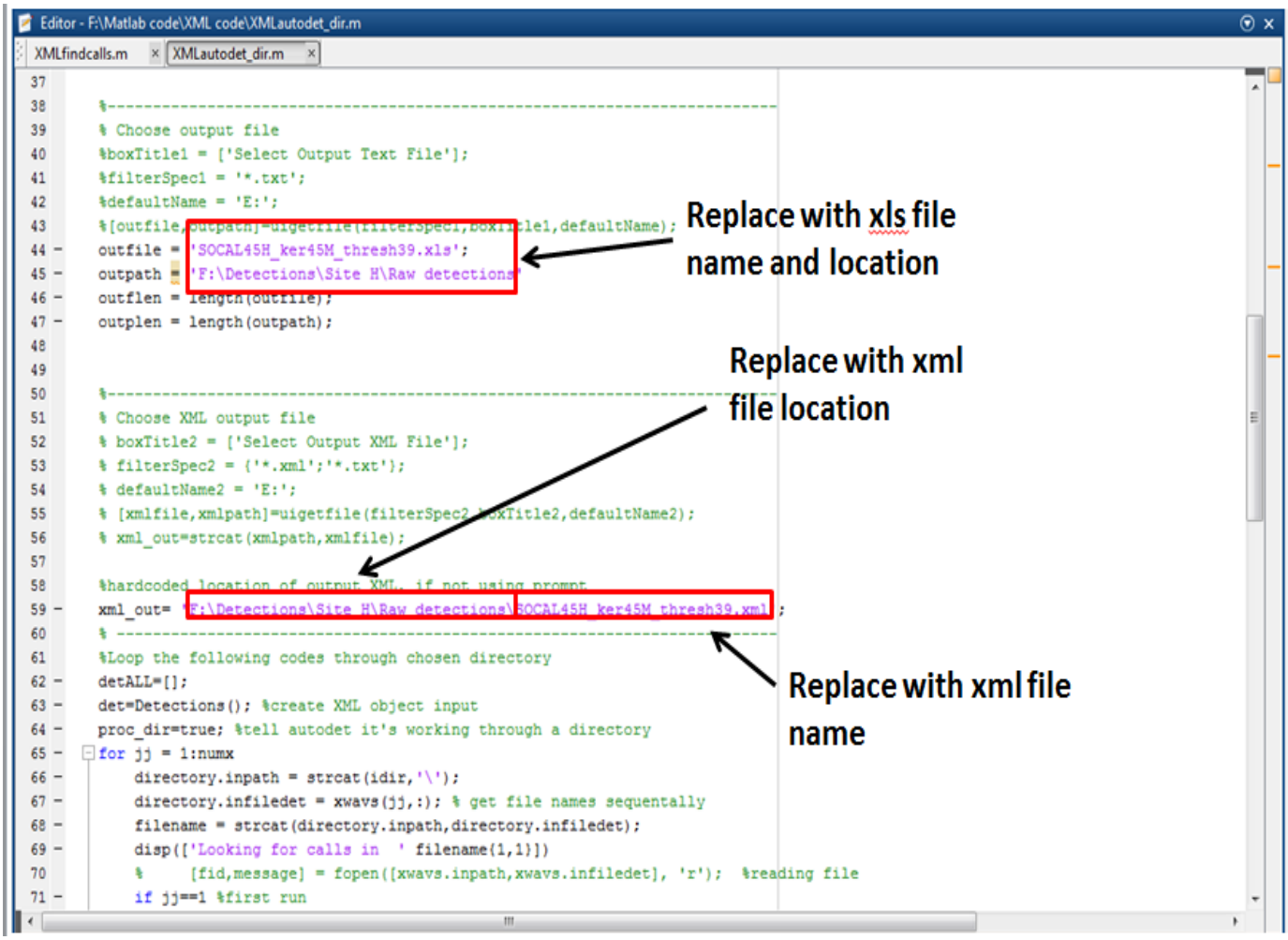


3. Click on XMLfindcalls.m and enter the correct kernel information (the same way you did when trying to determine the optimal threshold). Then scroll down and enter the threshold.

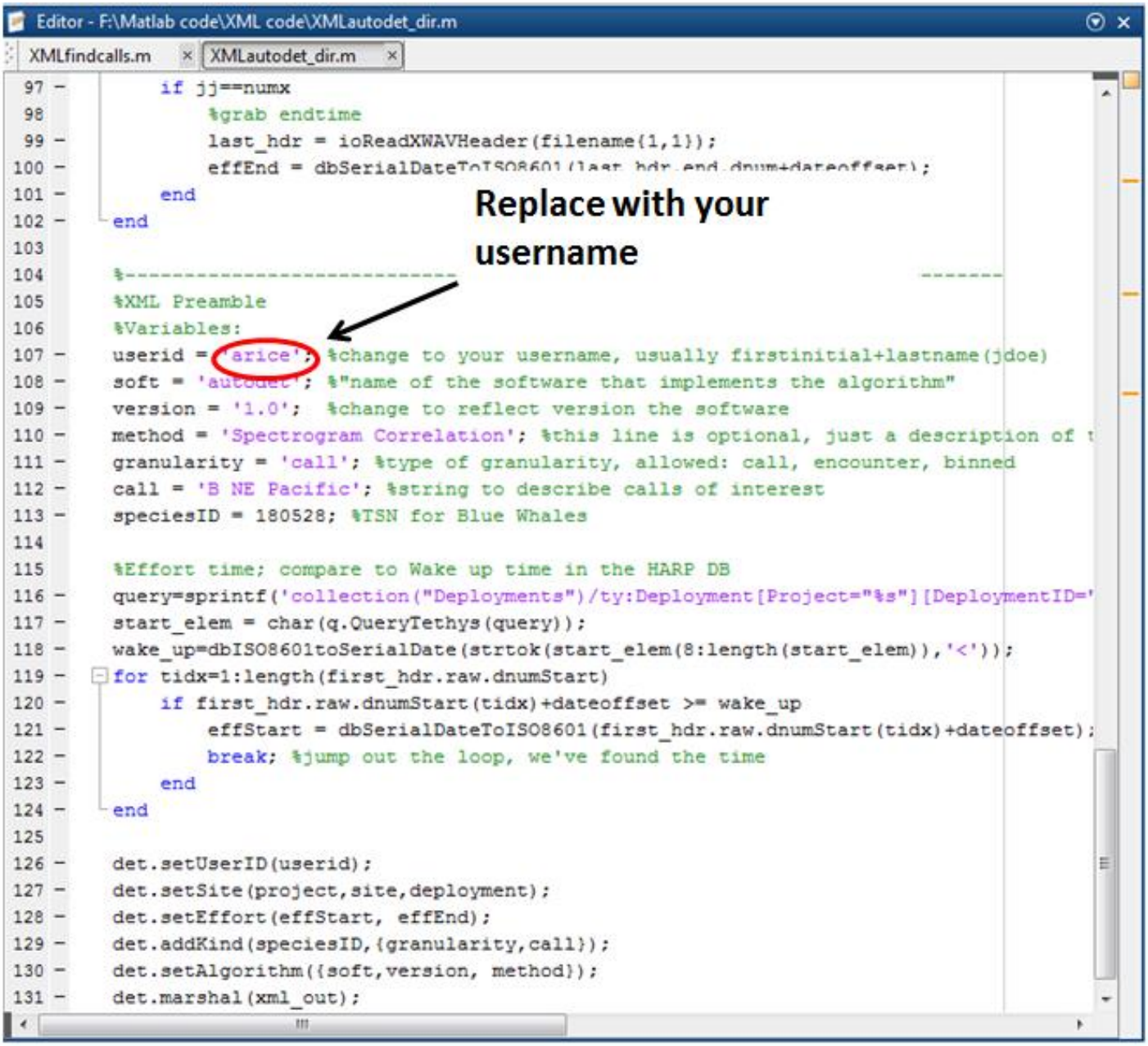


4. **SAVE** XMLfindcalls.m.

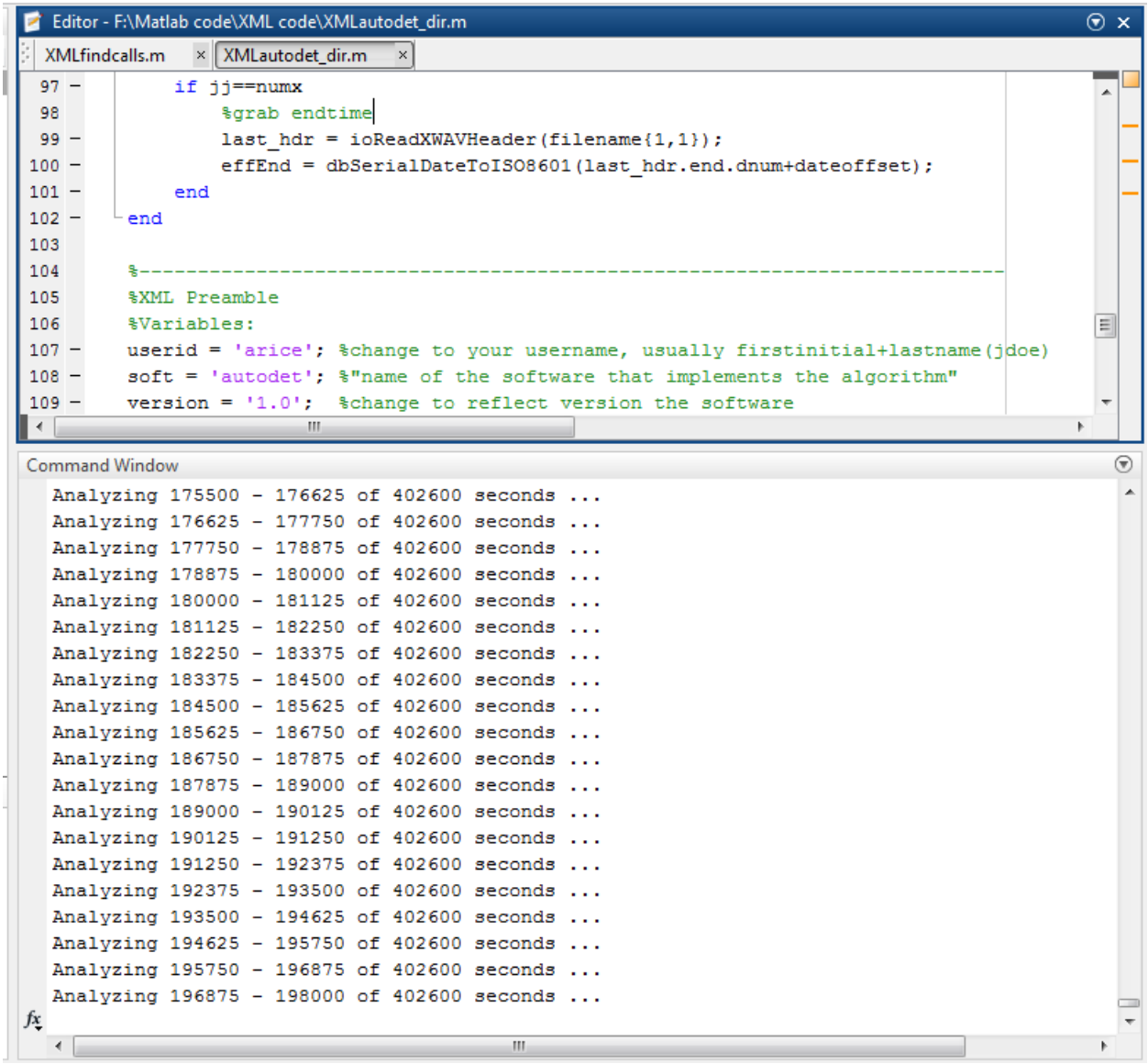
5. Click on XMLautodet_dir.m. Enter the name you would like the xls output file to be called (include the deployment info, the kernel name and the threshold) and below that enter the location where you would like the xls file to be placed. Then scroll down and enter the same information for the xml version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



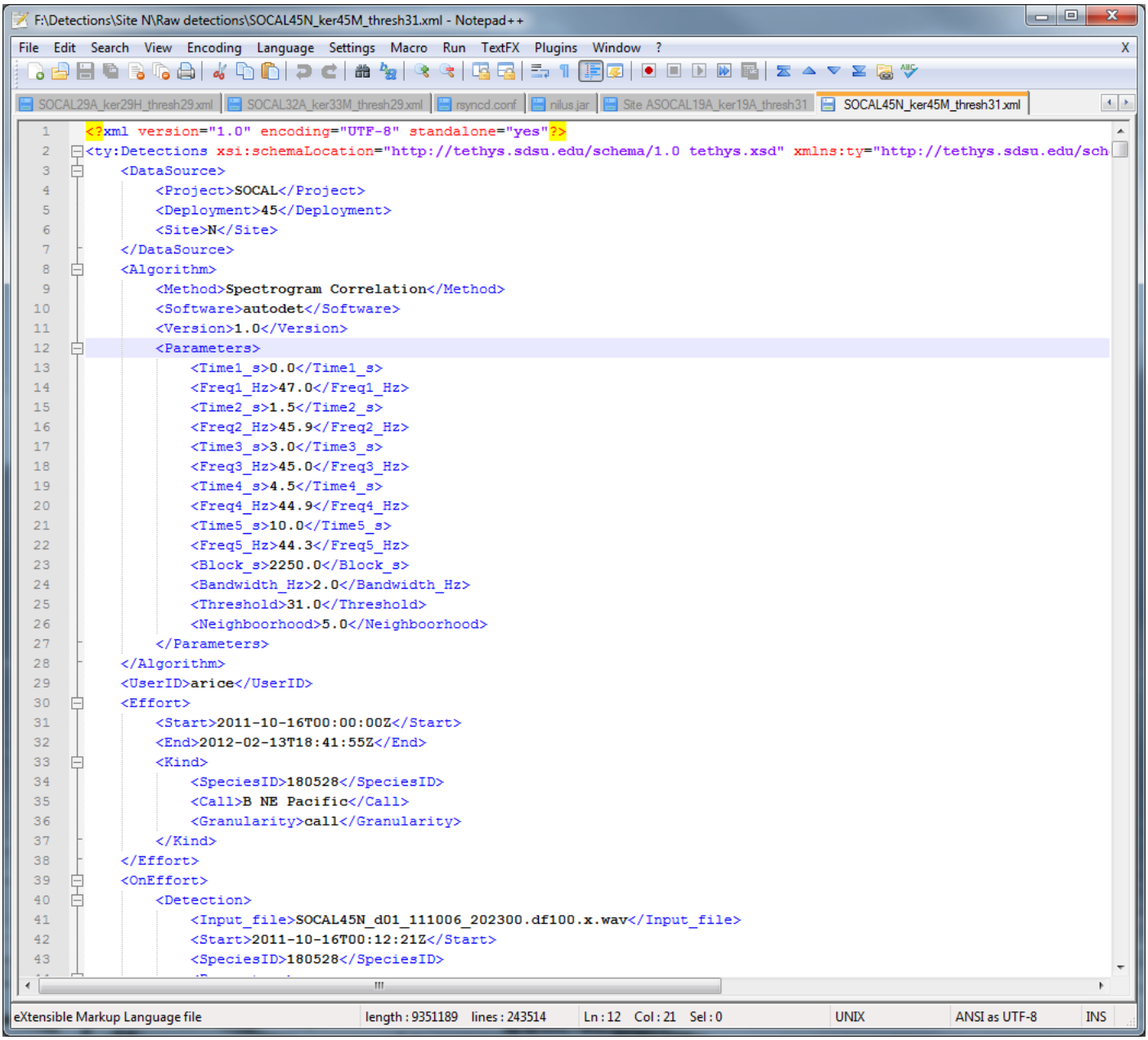
6. Then, scroll down further and change the current username to your username. You should only have to change the username once (when you save/run XMLautodet_dir.m your username will be saved along with it).



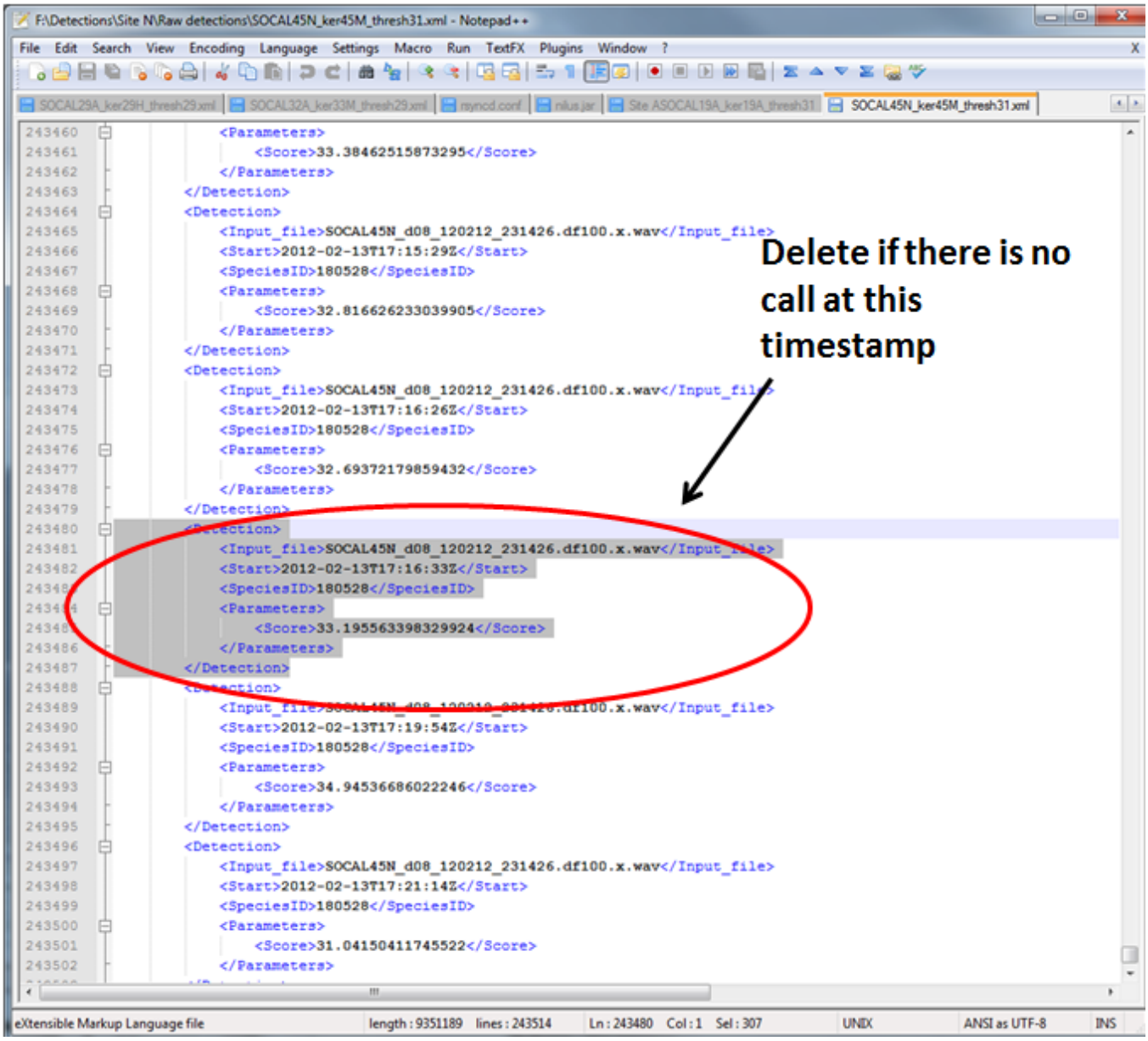
7. Run XMLautodet_dir.m. This will take a while.



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.



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.



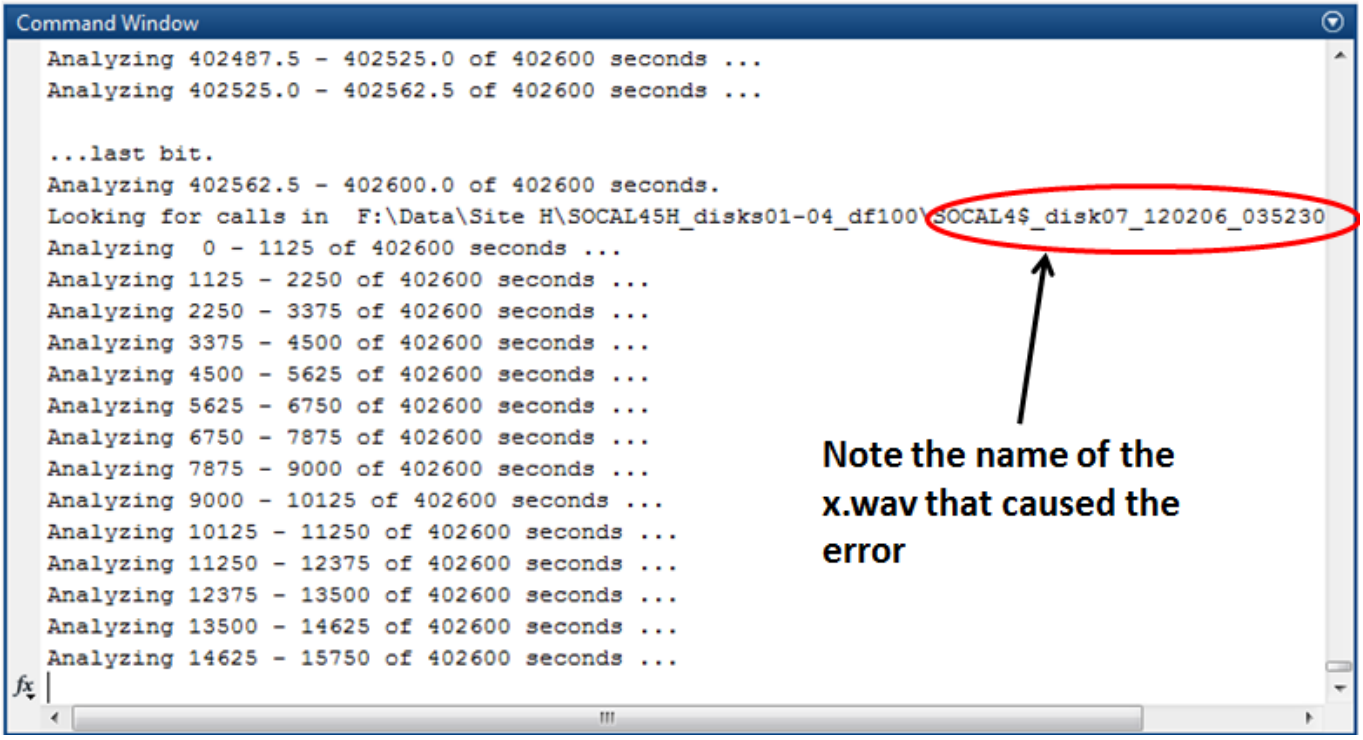
10. Whether or not you checked/deleted detections, **SAVE** the edited xml into a new folder (you don't want to overwrite the unedited detections). For each automatic detection you want to have a raw detection file and a checked detection file (representing what the detector gave you and what you're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

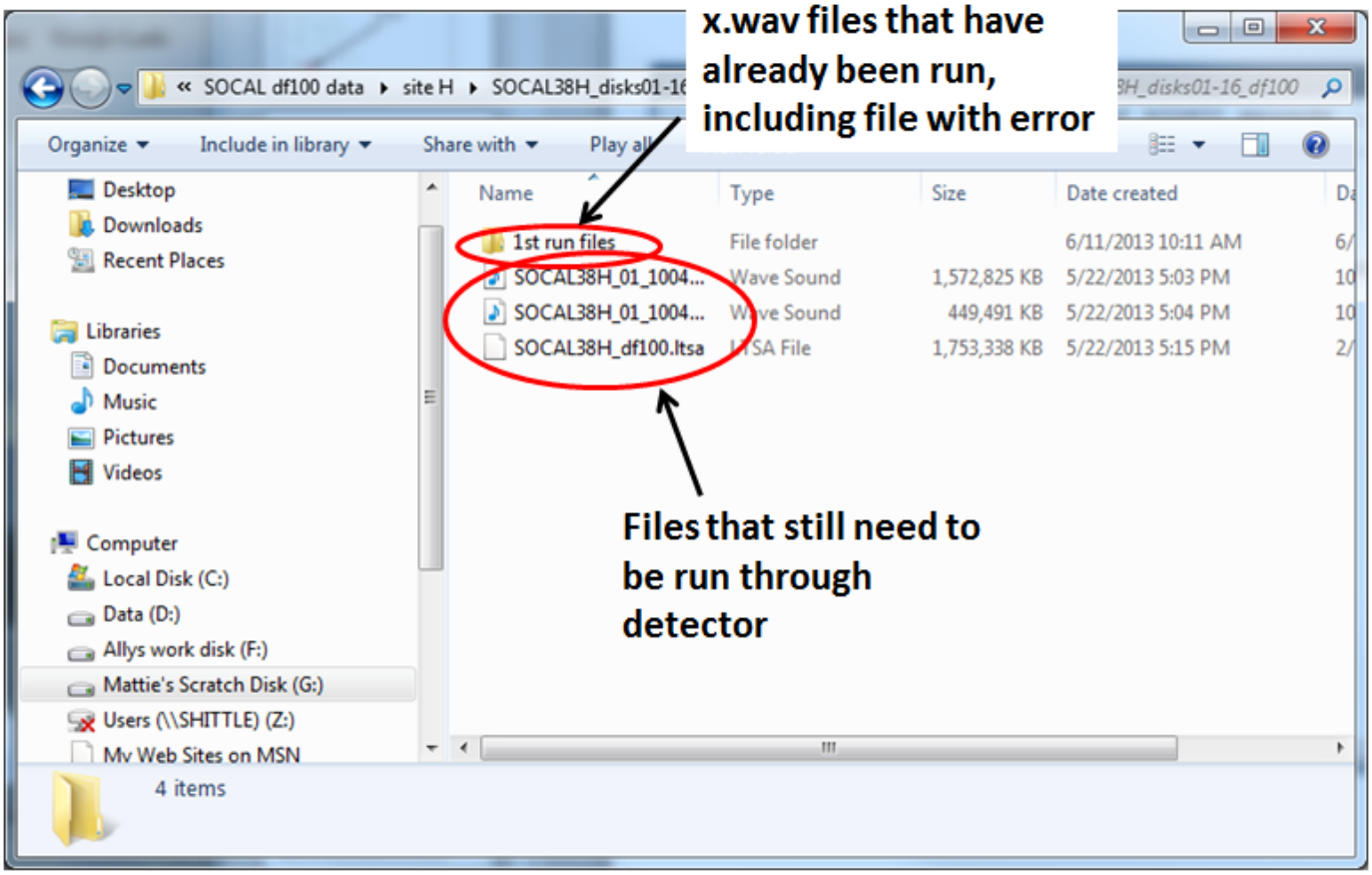
1. Make a note of which file it was that had the error.



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.