

Development of software for real-time acoustic identification of cetacean species

Final Report

PI: Whitlow Au, PhD

CoPI: Julie Oswald, PhD (Post-doctoral researcher)

Marine Mammal Research Program
Hawaii Institute of Marine Biology (HIMB)
46-007 Lilipuna Rd.
Kaneohe, HI 96744
Tel: (808) 247-5026; Fax: (808) 247-5831
E-mail: wau@hawaii.edu
oswald.jn@gmail.com



Introduction

Mitigation measures to reduce the risk of acoustic impacts of offshore oil and gas operations on marine mammals generally involve shipboard visual and acoustic monitoring. Visual detection and identification of cetaceans can be challenging as these animals spend much of their lives completely under water. In addition, visual surveys are labor intensive, costly, and contingent upon prevailing weather conditions, limiting them in the time span and geographic area that they can cover. Because many cetaceans produce characteristic calls that propagate well under water, passive acoustic techniques are becoming increasingly valuable in overcoming these limitations. The addition of towed hydrophone arrays to visual surveys makes it possible to detect groups of animals that would have been missed by visual observers due to animal behavior, weather, or visibility. Seafloor-mounted acoustic recorders can be deployed and left to record continuously for months at a time, allowing the collection of long time series of data that would be impossible to obtain using shipboard or aerial surveys alone.

While acoustic techniques have been shown to increase the rate and distance of cetacean detections during visual surveys (Thomas *et al.* 1986, Gordon *et al.* 2000), the ability to identify cetaceans to species using acoustic methods varies. Many large whales produce stereotyped calls that are easily recognized (ex. fin whales, Thompson *et al.* 1992; blue whales, Stafford *et al.* 1999). In contrast, the whistles produced by many delphinid species are highly variable and more difficult to identify (Matthews *et al.* 1999, Rendell *et al.* 1999). Nevertheless, time and frequency characteristics measured from spectrograms have been used to classify delphinid whistles to species in several studies (Steiner 1981, Matthews *et al.* 1999, Oswald *et al.* 2003). Correct classification scores obtained in these studies were significantly greater than expected by chance alone, suggesting that delphinid species can be identified based on characteristics of their whistles. Based on these results, Oswald (2006) developed Matlab-based software for automated acoustic species identification of delphinid whistles. This software, 'Real-time Odontocete Call Classification Algorithm' (ROCCA), automatically extracts whistle contours from sound files, measures ten variables from each contour, and applies classification algorithms (Discriminant Function Analysis, DFA, and Classification and Regression Tree Analysis, CART) to determine likely species identity. Initial tests of ROCCA resulted in schools of dolphins being correctly classified significantly more often than expected by chance ($p < 0.001$, Oswald 2006, Oswald *et al.* 2007), however correct classification was lower than the usual standards applied to visual identification during shipboard surveys (near certainty). With increased classification success, ROCCA will be a valuable tool, providing a method for identifying schools that are difficult to approach and observe, and allowing monitoring when visual efforts are compromised or unavailable. The inclusion of additional cetacean species would further increase the usefulness of ROCCA. In addition, ROCCA reduces processing time and will therefore make it feasible to run classification analyses on the large amounts of data collected using seafloor-mounted acoustic recorders.

The overall goals of this research project were to improve existing, Matlab-based software for the acoustic identification of marine mammal species (ROCCA) in the following ways:

- 1) increase classification success,
- 2) make the software more streamlined and user friendly,
- 3) expand its capabilities to include additional species and geographic locations,
- 4) convert ROCCA into the Java programming language and incorporate it into PAMGUARD

The following report summarizes the specific tasks involved in reaching each of these goals, the ways in which these tasks were approached, and the final outcome of each task.

Tasks

Task 1: Increase classification success

Subtask 1a: Explore other classification algorithms

The original version of ROCCA made classification decisions based on a combination of a 9-species discriminant function analysis (DFA) and a 9-species classification tree analysis (CART). Classification results are shown in Table 1. During the course of this research we investigated several different methods to improve classification success. The main methods explored are described below. All classifier work was done using a dataset of 2,231 whistles recorded in the tropical and temperate Pacific Ocean during six large scale marine mammal abundance surveys conducted by NOAA's Southwest Fisheries Science Center. This dataset includes the following seven species and one genus: common dolphins (*Delphinus* species), pilot whales (*Globicephala macrorhynchus*), false killer whales (*Pseudorca crassidens*), pantropical spotted dolphins (*Stenella attenuata*), rough-toothed dolphins (*Steno bredanensis*), striped dolphins (*Stenella coeruleoalba*), spinner dolphins (*Stenella longirostris*), and bottlenose dolphins (*Tursiops truncatus*). All recordings included in the analyses had visual confirmation of species identity and were localized acoustically to ensure that the animals being recorded were the same as those being observed visually. In addition, all were recordings made of single-species schools with no other species in the area.

The first method investigated for improving classification results involved a series of 2-class DFAs and CART analyses. Each of these 2-class classifiers classified one species against all other species (ex. bottlenose dolphin vs. everything else, spinner dolphin vs. everything else, etc.). The approach resulted in improvements for some species and worse results for others.

We then moved on to random forest analysis. A random forest is a collection of decision trees that are grown using binary partitioning of the data based on the value of one variable at each branch, or node. Randomness is injected into the tree-growing process by basing the decision of which variable to use as a splitter on a random subsample of the variables at each node (Breiman 2001). To test the random forest, the data were subsampled so that there were an equal number of whistles for each species. This avoided any one species swamping the data and skewing the results. Data were then randomly divided into two sets – a training dataset and a testing dataset. A random forest was created using the training dataset and classification results were determined by running the testing dataset through the random forest. Datasets were then switched so that each was used as a test and a train dataset and every whistle was classified. Based on

the results of these tests, it was determined that 500 trees provided the best trade-off between processing speed and correct classification scores.

One of the outputs of the random forest is a set of ‘variable importance scores’, which shows the degree to which each variable contributes to the random forest. Based on these scores, we tested different subsets of variables and found that using all 53 variables measured by ROCCA provided the best results.

The random forest analysis resulted in increases in correct classification scores for some species and decreases for others (see table 1). Overall, 43% of whistles were correctly classified using the Random Forest analysis, vs. 34% for DFA and CART. Even greater improvements were seen when the ‘strength’ of classifications were taken into account. The strength of a classification was measured based on the percent of trees within a random forest that ‘voted’ for the predicted species. A whistle was considered strongly classified if the predicted species received at least 35% of the votes, even if the prediction was incorrect. When classification decisions were based on strongly classified whistles alone, correct classification scores increased for most species, and overall 47% of whistles were correctly classified. (table 1). A threshold of 35% was determined based on the tradeoff between correct classification scores and the number of whistles that would be unclassifiable (since not all whistles were strongly classified). This threshold can be easily changed by users (see Appendix A - ROCCA Help File).

Table 1. Percent of whistles correctly classified by the original, Matlab version of ROCCA (‘Original ROCCA results’, Oswald et al. 2007) and by methods explored in during this research (‘New ROCCA results’).

Species	Original ROCCA results		New ROCCA results	
	DFA	CART	Random Forest	Random Forest – strong whistles (35% threshold)
Bottlenose	60.8	35.9	45.8	53.4
Spotted	29.6	33.8	45.8	44.0
Striped	14.7	26.2	45.8	55.6
Spinner	24.7	18.5	19.2	17.6
Rough-toothed	46.9	34.4	55.0	62.7
Common	28.3	21.7	35.8	36.8
False killer whale	22.6	57.1	65	70.7
Pilot whale	63.8	50.7	33.9	36.7
Overall	33.5	33.6	43.4	47.2

Another technique that was explored to increase the success of the random forest classifier was to create 'tiered' versions of ROCCA in which whistles were first classified to broad categories such as 'blackfish' or '*Stenella*' and then classified to species within the broad categories. No significant improvements were seen using this method and so the tiered version was not included in ROCCA.

Of the classification techniques that were explored during this research, basing classification decisions on strongly classified whistles using a random forest analysis produced the best results and so this is the method that is included in ROCCA.

ROCCA also allows the user to group individual whistle classifications into user-defined *schools*. ROCCA then classifies these schools to species by summing the number of trees voting for each species over all whistles within the school. The school is then classified as the species that the greatest number of trees voted for.

Subtask 1b: Create location-specific versions of ROCCA

Several different versions of ROCCA were created for specific geographic locations:

- 1) Tropical Pacific version: includes the following species recorded in the eastern tropical Pacific ocean and in the waters surrounding the Hawaiian islands:
-spinner dolphins, striped dolphins, spotted dolphins, short- and long-beaked common dolphins, bottlenose dolphins, rough-toothed dolphins, pilot whales, false killer whales
- 2) Temperate Pacific version: includes the following species recorded off the coasts of California, Oregon and Washington:
-short- and long-beaked common dolphins, bottlenose dolphins
- 3) Alaska version: includes transient killer whales (*Orcinus orca*) and beluga whales (*Delphinapterus leucas*) recorded in the Cook Inlet and Bering Sea.

In addition, a Chukchi Sea version of ROCCA is in progress. This version of ROCCA will include beluga whales, bowhead whales (*Balaena mysticetus*) and bearded seals (*Erignathus barbatus*) recorded in the Chukchi Sea. Work is expected to be completed on this classifier by the time the next version of PAMGUARD is released. This version of PAMGUARD will include ROCCA and all of its location-specific classifiers and is scheduled for release in late summer. The delay in the completion of the Chukchi classifier is due to the fact that the recordings were initially measured using methods different than those used by ROCCA. This could have affected classification results and therefore these recordings had to be re-analyzed.

Subtask 1c: Create tiered version of ROCCA

As described for Subtask 1a, several tiered versions of ROCCA were tested but none of these produced significant improvements so were not included in the final version.

Task 2: Automate call extraction

In the original Matlab-version of ROCCA, the extraction of the whistle contour involved several steps and could be quite time-consuming. This process has been modified significantly and the initial extraction of the tonal call now requires very little user-

interaction. The user simply has to click on the start and end points of the whistle and ROCCA extracts the whistle contour based on the frequencies of highest intensity in each time slice. This initial extraction is, however, not always very accurate and so several options exist for manipulating the extracted contour and increasing accuracy. These options include:

- 1) The ability to adjust the sensitivity to noise of the contour extraction algorithm
- 2) The ability to drag and drop contour points
- 3) The use of high- and low-pass filters to limit the extraction algorithm.

These contour adjustment options are described and illustrated in the *Contour Extraction/Manipulation* section of the ROCCA help file (Appendix A).

Task 3: Add species to ROCCA

As described for Subtask 1b, several species have been added to ROCCA, including: killer whales, beluga whales, bowhead whales and bearded seals. In addition, whistles recorded from species already included in ROCCA (short- and long-beaked common dolphins, bottlenose dolphins, pilot whales, false killer whales, rough-toothed dolphins, spinner dolphins, and striped dolphins) were also added. Although these species were already included in ROCCA, many of the additional recordings were collected in a different geographic location, allowing us to add a 'Temperate Pacific' classifier to ROCCA's suite of classifiers.

Tasks 4 and 5: Improve user interface and convert ROCCA to JAVA

The original, Matlab version of ROCCA required the user to input commands and had a very limited graphical user interface (GUI, Figure 1). The new PAMGUARD version of ROCCA is written in Java and has a complete GUI (Figure 2). All of ROCCA's functions are now controlled using easy to understand buttons and mouse-clicks. Functions of all of the buttons on ROCCA's GUI are described in the help files, Appendix A.

The Java-based, PAMGUARD version of ROCCA was beta-tested during a five month visual and acoustic marine mammal abundance survey that took place from August – December, 2010 in the Hawaiian Islands EEZ. During this survey, ROCCA's user interface, whistle extraction algorithm, and classification algorithm were all tested rigorously by experienced marine mammal acousticians. The acousticians provided regular updates and bug reports while at sea and a debrief meeting was held at the conclusion of the cruise. Bugs were fixed and ROCCA was improved based on feedback from the field test. This long-term, rigorous field testing resulted in a much more accurate, dependable and user-friendly tool for acoustic species identification.

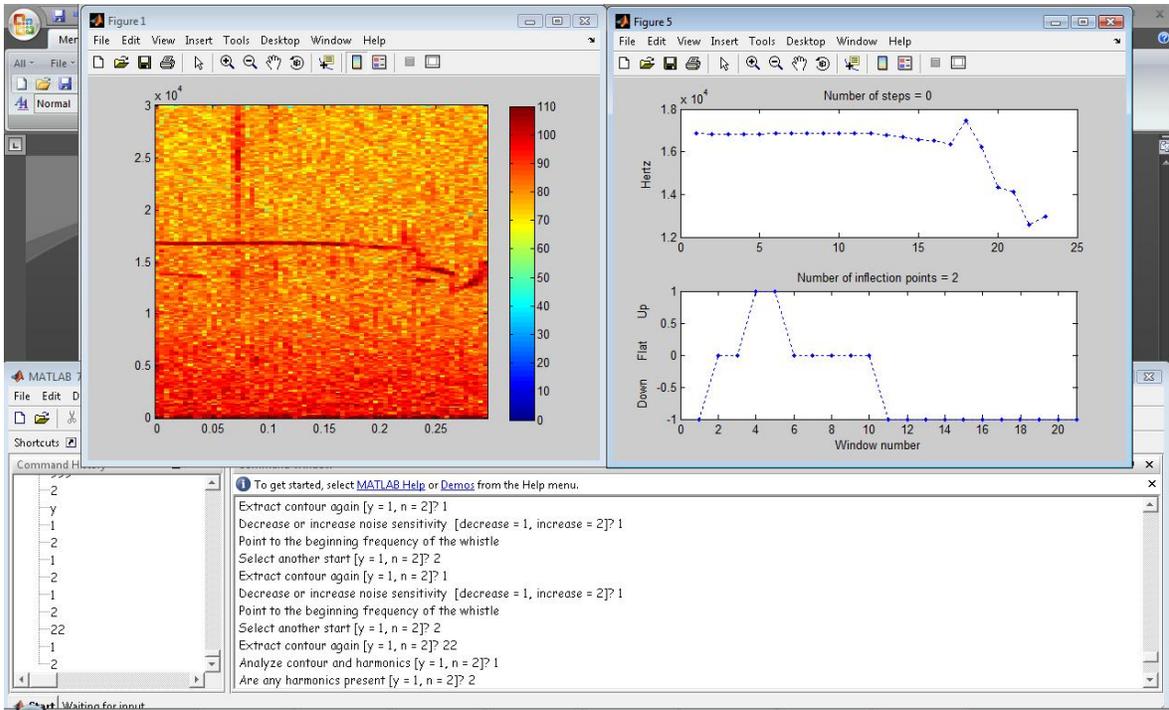


Figure 1. Command window and GUI for original, Matlab-based ROCCA.

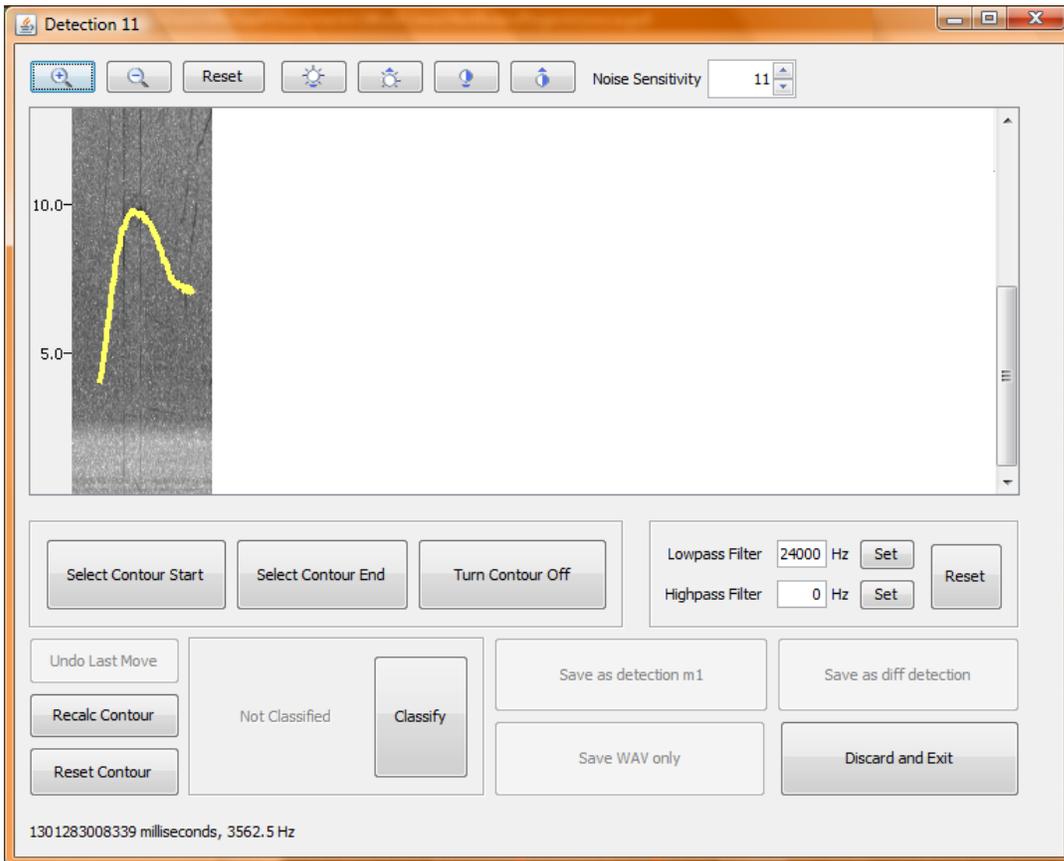


Figure 2. Graphical User Interface of PAMGUARDs ROCCA module.

Task 6: Publications

Two publications are in progress. The first, “Integration of ROCCA (Real-time Odontocete Call Classification Algorithm) into PAMGUARD signal processing software”, describes the optimized version of ROCCA and situations in which this software will be useful. It includes a description of all of the classifiers within ROCCA (tropical Pacific, temperate Pacific, Alaska, Chukchi). The second, “Seeing the species through the trees: using random forest classification trees to identify species-specific whistle types”, describes the random forest analysis and its performance in classifying delphinid whistles. Both of these papers were presented at the 161st Meeting of the Acoustical Society of America, Seattle, WA, 23-27 May 2011 and will be presented at the 5th International Workshop on Detection, Classification and Localization, Mount Hood, Oregon, 22-25 August, 2011. Abstracts of all of these presentations are included in Appendix B. We anticipate submitting both of these publications to the Journal of the Acoustical Society of America by the end of summer 2011.

Task 7: Users Manual

A user’s manual, in the form of help files has been written in HTML, to properly interface with the Pamguard help system. These help files explain all of the functions available in ROCCA, as well as how to set up the different classifiers. These help files are included as Appendix A in this report.

Literature Cited

- Breiman, L. 2001. “Random Forests,” *Machine Learning* 45:5-32.
- Gordon, J.C.D., J.N. Matthews, S. Panigada, A. Gannier, J.F. Borsani, and G. Notarbartolo di Sciara. 2000. Distribution and relative abundance of striped dolphins in the Ligurian Sea Cetacean Sanctuary: Results from an acoustic collaboration. *J. Cet. Res.* 2:27–36.
- Matthews, J.N., L.E. Rendell, J.C.D. Gordon, and D.W. MacDonald. 1999. A review of frequency and time parameters of cetacean tonal calls. *Bioacoustics* 10:47–71.
- Oswald, J.N. 2006. An examination of the whistling behavior of small odontocetes and the development of methods for species identification of delphinid whistles. Ph.D. dissertation, University of California, San Diego, CA.
- Oswald, J.N., J. Barlow, and T.F. Norris. 2003. Acoustic identification of nine delphinid species in the eastern tropical Pacific ocean. *Mar. Mamm. Sci.* 19:20–37.
- Oswald, J.N., S. Rankin, J. Barlow, and M. Lammers. 2007. A tool for real-time acoustic species identification of delphinid whistles. *J. Acoust. Soc. Am.* 122:587-595.
- Rendell, L.E., J.N. Matthews, A. Gill, J.C.D. Gordon, and D.W. MacDonald. 1999. Quantitative analysis of tonal calls from five odontocete species, examining interspecific and intraspecific variation. *J. Zool.* 249:403–410.
- Stafford, K.M., S.L. Nieukirk, and C.G. Fox. 1999. An acoustic link between blue whales in the eastern tropical Pacific and the northeast Pacific. *Mar. Mamm. Sci.* 15:1258–1268.
- Steiner, W.W. 1981. Species-specific differences in pure tonal whistle vocalizations of five western north Atlantic dolphin species. *Behav. Ecol. Sociobiol.* 9:241–246.
- Thomas, J.A., S.A. Fisher, and L.M. Fern. 1986. Acoustic detection of cetaceans using a

towed array of hydrophones. Rept. Int. Whal. Commn. (Spec. Iss. 8):139–148.

Thompson, P., L.T. Findley, and O. Vidal. 1992. 20 Hz pulses and other vocalizations of fin whales, *Balaenoptera physalus*, in the Gulf of California, Mexico. J. Acoust. Soc. Am. 92:3051–3057.

Appendix A – Rocca Help File

Real-time Odontocete Call Classification Algorithm

A tool for real-time acoustic species
identification of delphinid whistles

Users Guide v.1b
July 2011

For further information, please contact Dr. Julie Oswald at oswald.jn@gmail.com

Rocca

Overview

The ROCCA (Real-time Odontocete Call Classification Algorithm) module classifies delphinid whistles based on spectrographic measurements extracted from whistle contours. The contour extraction algorithm [measures](#) 54 different variables for each contour. A subset of these variables is then used in conjunction with a user-selected classification model to classify the whistle to species.

Individual whistle classifications are grouped into user-defined [Detection Numbers](#). A classification is also given to detection numbers based on the species within that group with the [highest classification scores](#).

General Principle of Operation

There are 5 main steps to the classification of a whistle:

1. User selects a whistle from the spectrogram display
2. The whistle is captured and displayed in a new spectrogram window
3. The whistle contour is extracted and the user is allowed to manipulate it, if desired
4. The contour parameters are measured
5. The contour is classified using the currently-loaded classifier model
6. The classified whistle statistics are added to the specified School/Detection/Sighting

[Next: Configuring Rocca](#)

Rocca

Creation and Configuration

Creating a Rocca module

From the **File > Add Modules > Detectors** menu, select "Rocca". Enter a descriptive name for the new classifier and press OK.

Rocca requires an [FFT engine module](#) to operate. If no FFT engine module has been loaded, the user will be prompted to load one.

When the module is created, Rocca will automatically try to load the [classifier model](#). Rocca will also check to see if the [School Stats file](#) already exists. If it does, the user is given the option to load the contents of the file into memory.

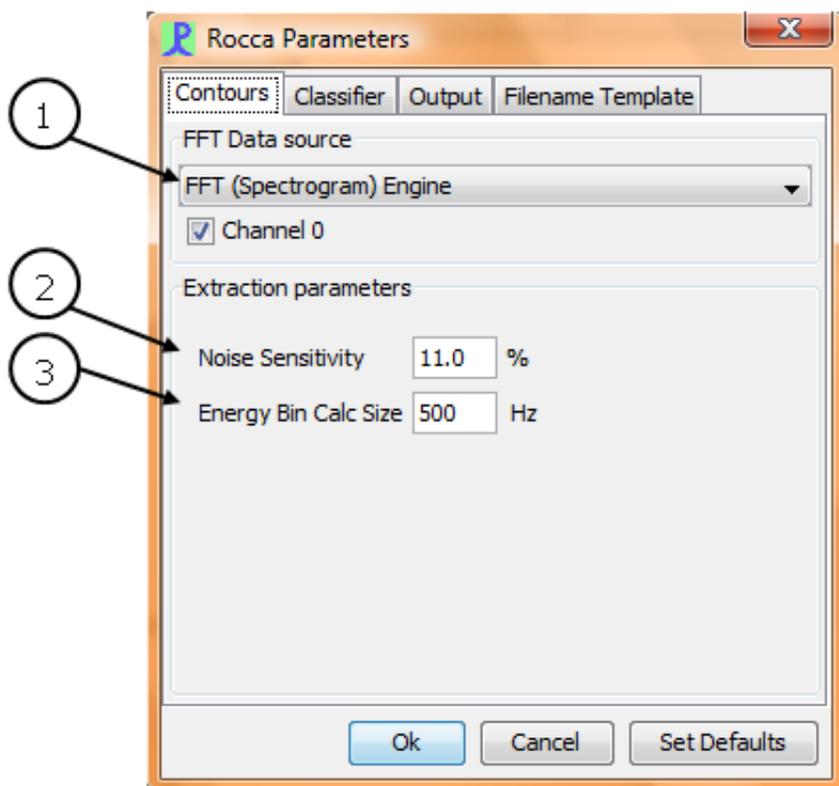
Note: the School Stats file is continually overwritten while Rocca is running. If, upon startup, the user does not load the data into memory, the information in the file will be lost. Thus if the user does not wish to load the data *but* still desires to keep it, the file should be renamed before continuing with Pamguard.

Configuring a Rocca module

Note: it is recommended that the [Sound Acquisition](#) and [FFT Engine](#) modules be configured before configuring Rocca.

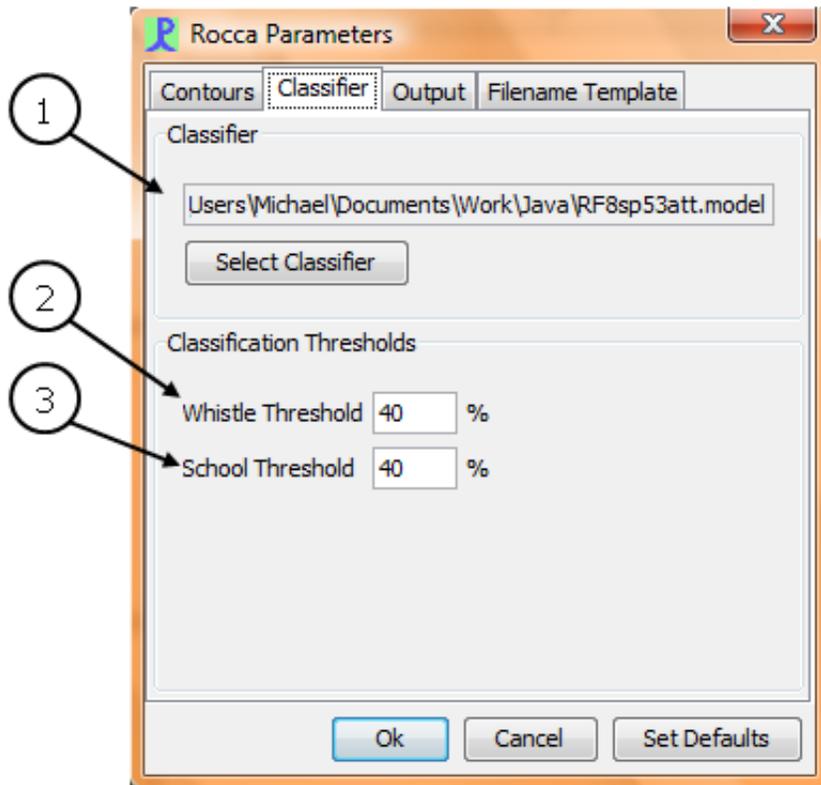
From the Detection menu, select "Rocca Parameters". A new dialog should appear containing four tabs, [Contour](#), [Classifier](#), [Output](#), and [Filename Template](#).

Contour tab



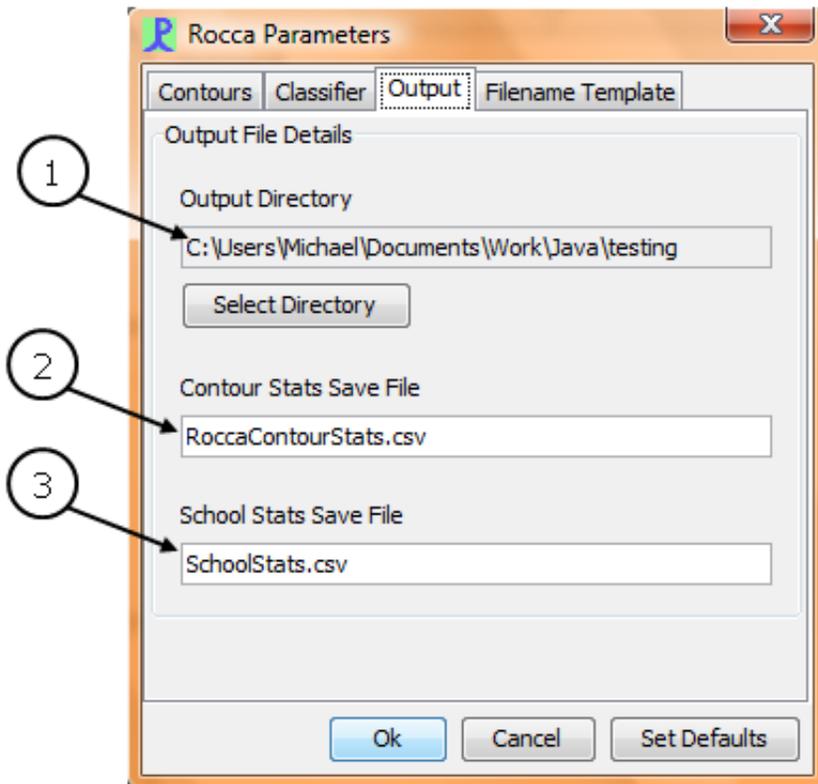
1. FFT Data source: the fft data source to be used. The channels listed are those currently selected in the [FFT Parameters dialog](#). Rocca will save a wav clip of a whistle contour whenever it is classified and added to the current detection. Check the channel boxes corresponding to the channels you wish to save to this clip file. Note that this is different than the channels displayed in the spectrogram window, which are set in the [display options dialog](#). Whistles can be selected and contours extracted from any channel displayed, but the saved whistle clips will only contain data from the channels selected here. So it is possible (though not recommended) to extract and classify a whistle from Channel 0, but only save the data from Channels 1 and 2.
2. Noise Sensitivity: specify the global noise sensitivity parameter to use when [extracting a contour](#).
3. Energy Bin Calc Size: specify the size of the frequency bin used to calculate the energy around each peak frequency.

Classifier tab



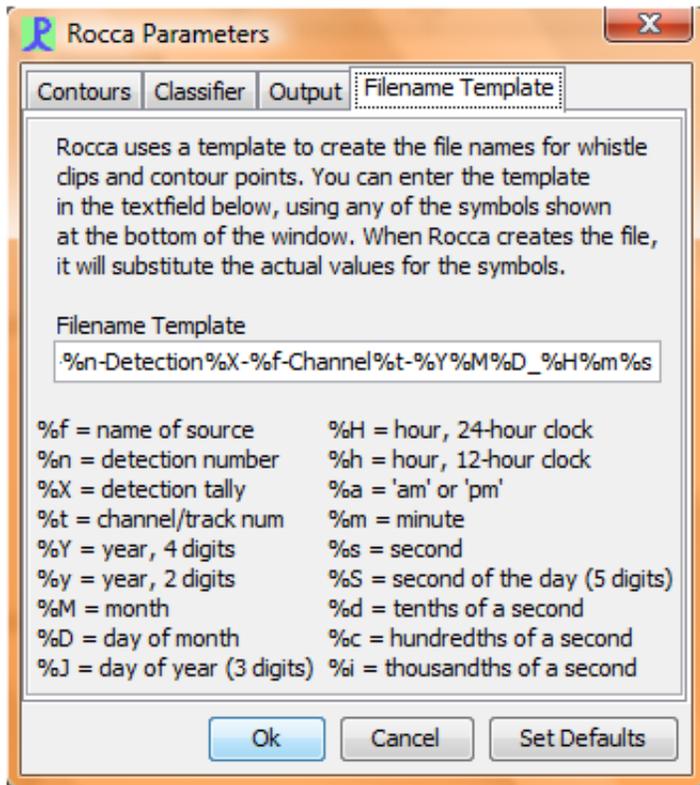
1. Classifier: select the classifier to be used. Classifier models are created based on the [Weka](#) Random Forest model. A classifier model uses the file extension .model. At the moment the ability to create new classifiers within Pamguard is not available. For a new classifier based on custom data, please contact Dr. Julie Oswald at oswald.jn@gmail.com
2. Whistle Threshold: the threshold value to use when [classifying individual whistles](#)
3. School Threshold: the threshold value to use when [classifying detection numbers](#)

Output tab



1. **Output Directory:** specify the directory to [save](#) clip files, whistle contour points and whistle contour parameters to. Note that, with Microsoft Windows operating systems, it is a good idea to select a directory that does not require administrator privileges to save to.
2. **[Contour Stats Save File](#):** specify the name of the file to save classified whistle contour parameters to. The file will be saved to the output directory, specified above. Each whistle classified is appended to the end of the file when one of the Save buttons in the Spectrogram Popup window is clicked.
3. **[School Stats Save File](#):** specify the name of the file to save the school classifications to. The file will be saved to the output directory, specified above. Note that in order to use the latest detection numbers, this file is overwritten during each save. While this is not a problem as long as Pamguard is run continuously, if this file exists when Pamguard is first loaded the contents will be lost. If the file exists during startup, the user is warned and given a chance to load/rename/backup the existing file.

Filename Template tab



In addition to the Contour Stats and School Stats save files, Rocca also saves a whistle clip (wav format) and a list of the contour points (time and frequency, csv format). These files will be named according to the template given in the text box on this tab. As part of the name, the user is able to use any of the following symbols. When the files are saved, Rocca will substitute the actual values for the symbol names.

Symbol	Meaning	Symbol	Meaning
f	name of source	H	hour, 24-hour clock
n	detection number	h	hour, 12-hour clock
X	detection tally/count	a	<i>am or pm</i>
t	channel/track number	m	minute
Y	year, 4 digits	s	second
y	year, 2 digits	S	second of the day, 5 digits
M	month	d	tenths of a second
D	day of the month	c	hundredths of a second
J	day of the year, 3 digits	i	thousandths of a second

Note that all date/time values are GMT.

Rocca Mark Observers

Rocca must be added to the [Mark Observers](#) list in the spectrogram to be monitored. Right click on the spectrogram and select Settings. Select the Mark Observers tab and check the box beside Rocca. If Rocca

is not listed as a possible mark observer, it has not been added as a module.

[Previous: Rocca Overview](#)

[Next: Selecting a Whistle](#)

Rocca Selecting a Whistle

To classify a whistle using Rocca, the user must first select the whistle from the scrolling spectrogram display. Move the cursor to one corner of the whistle, press and hold the mouse button while dragging the cursor to the opposite corner, and release the button. A red box will be drawn around the selection, and the time and frequency of the starting point (lower left corner) and ending point (upper right corner) are displayed in the Rocca sidebar for reference.



When the mouse button is released, a new window will be opened containing that portion of the spectrogram selected by the user. The new window is minimized automatically.

If a [Detection Number](#) has not yet been specified, the user will be prompted for one first.

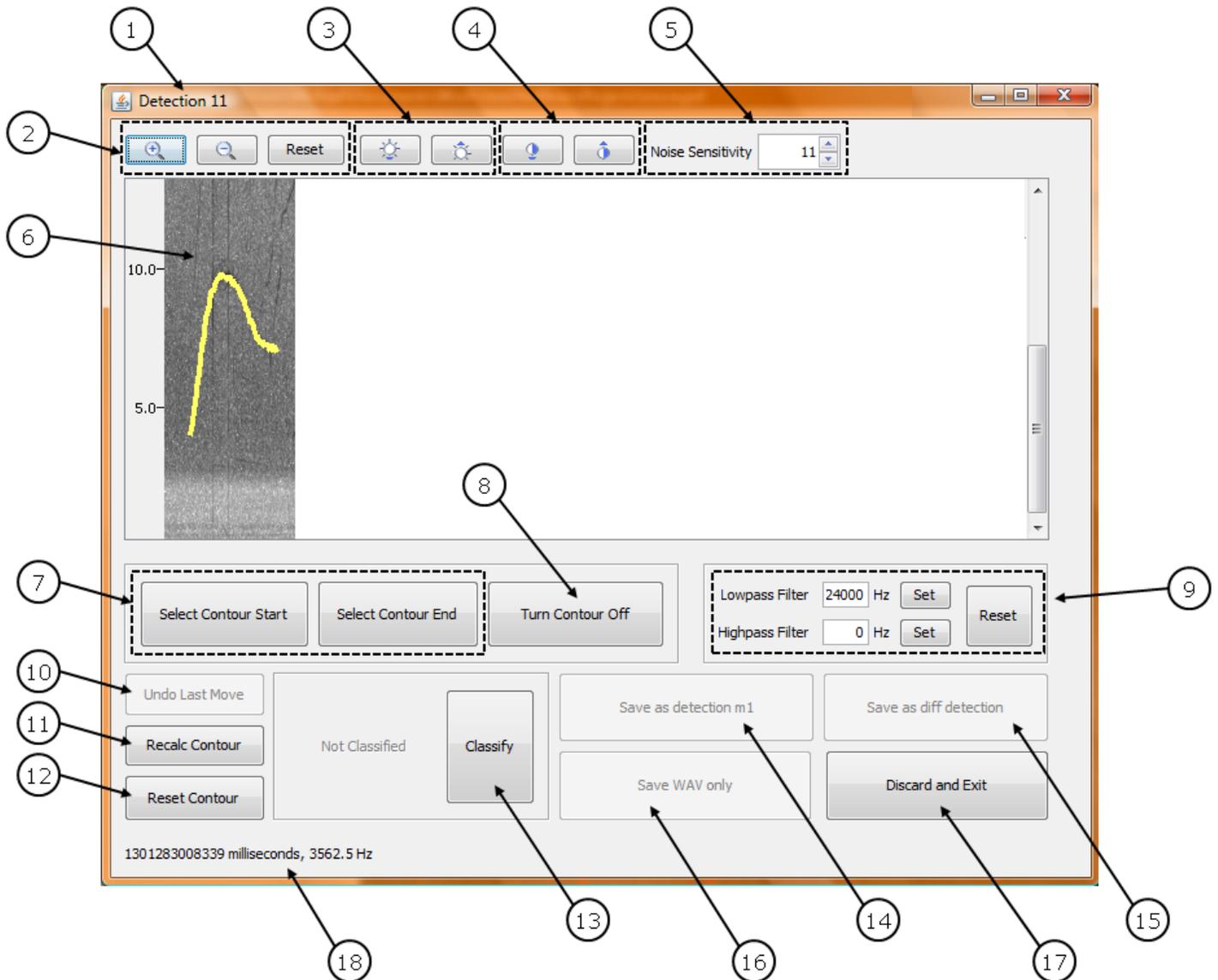
At this stage, the whistle of interest has been captured and the user is able to continue monitoring the scrolling spectrogram for additional acoustic events. The user does NOT need to classify the whistle immediately. The number of captured whistles is only limited by the amount of memory available on the computer.

[Previous: Rocca Creation and Configuration](#)

[Next: Spectrogram Popup Window](#)

Rocca Spectrogram Popup Window

Once a whistle has been selected in the main spectrogram window, a new window opens containing the portion of the spectrogram selected by the user. The user will need to click on the starting point of the whistle, and then the ending point (note that these can be reselected later). Once the starting and ending points have been defined, Rocca will extract and display the whistle contour (shown in yellow).



1. Window Title: displays the current detection tally. Rocca keeps track of how many detections have been made since Pamguard has been started and displays the information in the window title to make it easier to keep track of detections during an encounter with multiple acoustic events.
2. Zoom In/Out/Reset: changes the zoom level of the spectrogram.
3. Increase/Decrease Brightness: modifies the brightness of the spectrogram. Note that this only modifies the spectrogram display â€” the underlying contour extraction algorithm is not affected.
4. Increase/Decrease Contrast: modifies the contrast of the spectrogram. Note that this only modifies the spectrogram display â€” the underlying contour extraction algorithm is not affected.
5. Noise Sensitivity spinner: modifies the noise sensitivity parameter used for [contour extraction](#). The contour is automatically recalculated when this value is changed, and the spectrogram display is updated. Note that any contour points [manually moved](#) will not be changed. Note also that this does not change the global noise sensitivity parameter as specified in the [Rocca Parameters window](#), only the noise sensitivity used in this spectrogram popup window.
6. Spectrogram window: the current spectrogram.
7. Select Contour Start/Contour End: allows user to reselect the starting or ending point of the whistle. Upon selection, the contour will recalculated automatically.

8. Turn Contour Off: toggle the contour display on and off.
9. Highpass/Lowpass/Reset filters: limits the [contour extraction algorithm](#) to only function within a specific frequency band. Specify a filter by either typing in the desired frequency and hitting <enter>, or hitting the <Set> button and clicking on the desired frequency in the spectrogram. The frequency filter is drawn as a horizontal red line on the spectrogram. The contour is automatically recalculated when these values are modified, and the spectrogram display is updated. Reset the filters by clicking the <Reset Filters> button. Note that any contour points [manually moved](#) will not be changed. Note also that the spectrogram display is not affected by the filters, only the contour extraction.
10. Undo Last Move: return the last contour point that was [moved manually](#) to its previous location.
11. Recalc Contour: recalculate the contour after [manually moving](#) one or more contour points to new positions.
12. Reset Contour: unlock all manually moved contour points.
13. Classify/Reclassify: classify the currently-extracted contour. The classification is displayed to the left of the button.
14. Save as detection *det#*: save the whistle clip, contour points, and contour parameters to the directory specified in the [Rocca Parameters window](#) using *det#* as the [detection number](#). Close the spectrogram popup window and return to the main Pamguard display. Classification results will be added to the Rocca [Sidebar](#).
15. Save as diff detection: save the whistle clip, contour points, and contour parameters to the directory specified in the [Rocca Parameters window](#), but first query the user as to which [detection number](#) to use. Close the spectrogram popup window and return to the main Pamguard display. Classification results will be added to the Rocca [Sidebar](#).
16. Save WAV only: save the whistle clip to the directory specified in the [Rocca Parameters window](#) using the current detection number (shown in the window title bar). This option is useful if the user wishes to classify the whistle later.
17. Discard and Exit: discard the current whistle, close the spectrogram popup window and return to the main Pamguard display.
18. Displays the current time and frequency location of the cursor.

[Previous: Selecting a Whistle](#)

[Next: Contour Extraction / Manipulation](#)

Rocca

Contour Extraction / Manipulation

The whistle contour is automatically extracted from the spectrogram by stepping through the spectra one time slice at a time and measuring the peak frequency for each slice within a specific frequency bin. The upper and lower limits of the search bin are defined by the peak frequency of the previous time slice +/- the noise sensitivity (as defined in the [Rocca Parameters window](#)). The frequency bin of the first time slice is determined by the user-selected start frequency +/- the noise sensitivity.

Adjusting the Noise Sensitivity

If the contour does not match the underlying whistle, the first step should be to adjust the noise sensitivity.

- Decreasing the sensitivity narrows the frequency band used when searching for the peak frequency in the next time slice. A smaller band means the extraction algorithm is less likely to jump from the whistle contour to a nearby noisy peak. However, a smaller band also means that if the algorithm does jump to a nearby noisy peak, there is less chance it will be able to jump back to the whistle contour.
- Increasing the sensitivity widens the frequency band used when searching for the peak frequency in the next time slice. A wider band increases the chances that an extraction gone off-track and following noise will be able to jump back to the whistle contour. However, a wider band also increases the chances of going off-track in the first place.



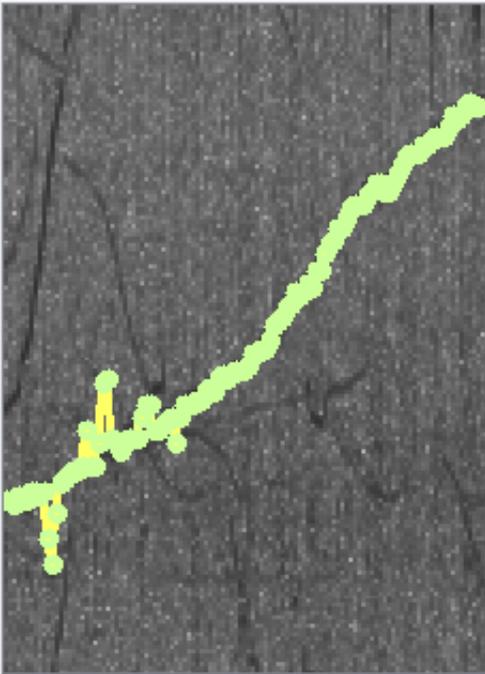
A button is provided in the top toolbar to quickly adjust noise sensitivity. The spectrogram will automatically update with the recalculated contour each time the sensitivity is changed.

Adjusting the highpass/lowpass filters

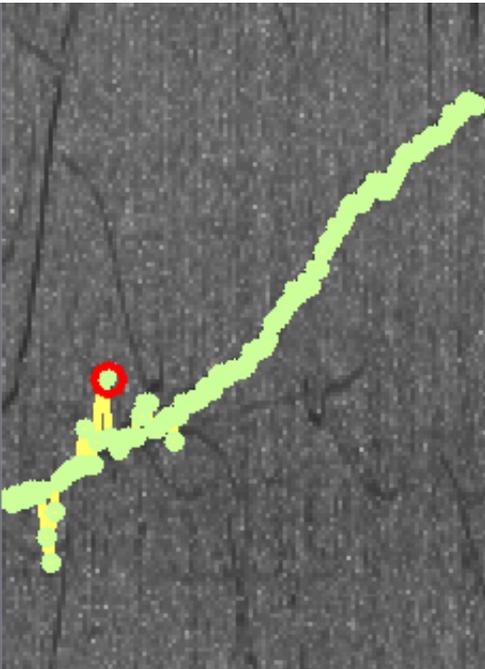
The highpass and lowpass filters limit the contour extraction algorithm to only function within a specific frequency band. The highpass filter defines the lower bound, and the lowpass filter defines the upper bound. Specify a filter by typing in the desired frequency and hitting <enter>, or hitting the <Set> button and clicking on the desired frequency in the spectrogram. The filter is displayed on the spectrogram as a red horizontal line. If the extraction algorithm finds a peak frequency lower than the highpass filter specified, the peak frequency will be set to the value of the highpass filter. Similarly, if the extraction algorithm finds a peak frequency higher than the lowpass filter specified, the peak frequency will be set to the value of the lowpass filter.

The filters can be reset by clicking the [<Reset Filters> button](#).

Adjusting the contour points manually



Individual contour points can be dragged manually to new positions. In order to view the points, move the mouse over the spectrogram and press the mouse button. Try to move the mouse to a position away from the current whistle contour, to avoid accidentally dragging a contour point. Contour points will be displayed as light green dots.



Contour points can only be dragged vertically. That is, the frequency can be changed but not the time slice. Once the contour point to be moved is determined, position the mouse over top of the point and press and hold the mouse button. The contour points will again be displayed as light green dots, but the point closest to the cursor will have a red circle. Move the mouse up or down to the desired frequency, and release the mouse button.

Once a contour point has been moved, it will be locked in the new position even if the contour is recalculated by adjusting the noise sensitivity, modifying filters or clicking the [<Recalc Contour> button](#). In this way, if the contour extraction algorithm has followed noise instead of the desired whistle, the user can move a point to the correct position and get the extraction back on track. The contour point can still be moved manually, and clicking the [<Reset Contour> button](#) will unlock all locked points.

[Previous: Spectrogram Popup Window](#)

[Next: Whistle and School Classification](#)

Rocca

Whistle and School Classification

Rocca currently uses a Random Forest classifier model based on the open-source statistical software package [Weka](#). For more information on Random Forests and the WEKA package, the user is encouraged to refer to the book [Data Mining: Practical Machine Learning Tools and Techniques](#)

Whistle Classification

The following 54 parameters are measured for each whistle contour:

minimum frequency, maximum frequency, duration, beginning frequency, ending frequency, median frequency, center frequency, frequency range and relative bandwidth, mean and standard deviation of the frequency, mean and the standard deviation of the duty cycle, the frequency max/min ratio, the frequency beginning/ending ratio, the frequency at the 25th, 50th and 75th percentiles and the spread between the 75th and 25th percentiles, the mean duty cycles in the first, second, third and fourth quarters of the whistle, the coefficient of frequency modulation, the number of frequency steps up and down, the total number of steps, the mean slope, the mean absolute slope, the mean positive and negative slopes, the ratio of mean positive slope to mean negative slope, the slope (up/down/flat) at the beginning and ending of the contour, positive and negative boolean parameters describing the slope at the beginning and ending of the contour, the percentage of upsweeps, downsweeps and no sweeps (flat), the number of sweep patterns (up-down, down-up, up-flat, down-flat, flat-up, flat-down), the number of inflection points, the shortest and longest times between inflection points and the ratio of longest to shortest, the mean and standard deviation of the time between inflection points, and the median time between inflection points.

The random forest classifier is created based on a subset of these parameters. During whistle classification, the parameters from the whistle contour are dropped down the random forest model and each possible species is assigned a percentage of the votes. The whistle is classified as the species with the highest percentage of votes, unless that percentage falls below the whistle threshold (as specified in the [Rocca Parameters window](#)). In that instance, the whistle is classified as Ambiguous.

School Classification

Each [detection number](#) holds a list of possible species based on the classifier model used. There are two values stored for each species: the number of times a whistle has been classified to that species (displayed), and a cumulative total of all the percentage tree votes for the species (not displayed). So when a new whistle classification is saved to a detection number, the count of the classified species is increased by one and the percentage tree votes for each species are added to the corresponding cumulative totals.

The detection is classified as the species with the highest cumulative percentage of votes. Note that this may be different than the species most often classified - the value shown in the sidebar species list. If the highest cumulative percentage falls below the school threshold (as specified in the [Rocca Parameters window](#)), the detection is classified as Ambiguous.

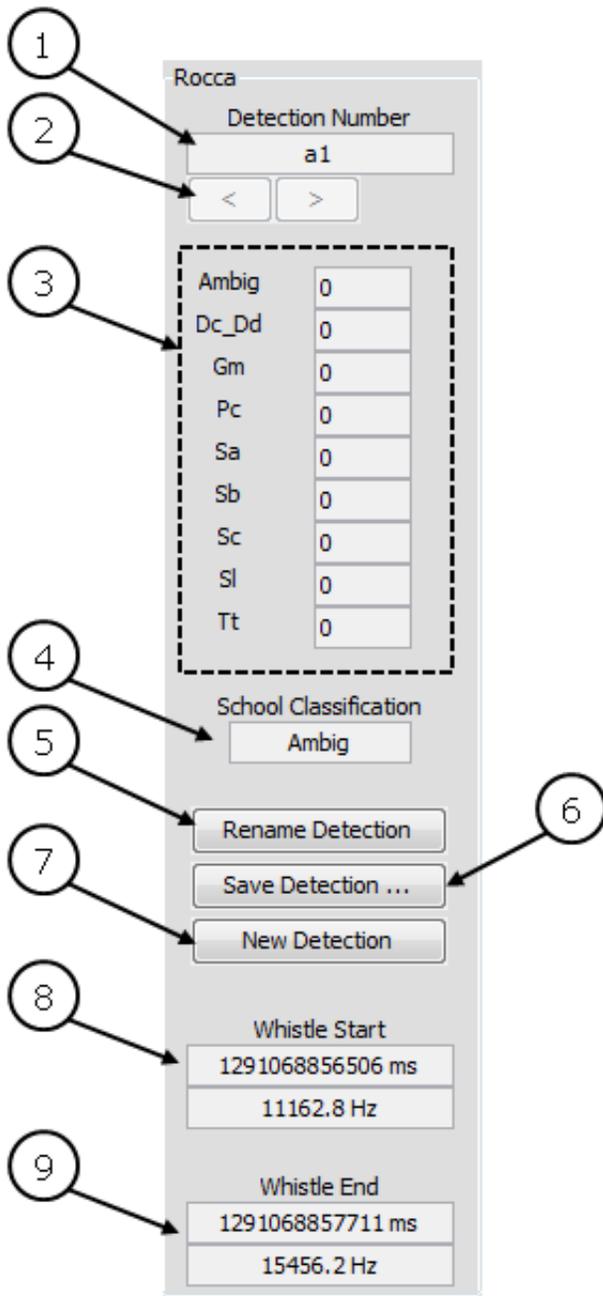
[Previous: Contour Extraction / Manipulation](#)

[Next: Rocca Sidebar](#)

Rocca

Rocca Sidebar

Rocca assembles the results of individual whistle classifications into groups as defined by the user. Each group must be given a name, the **Detection Number**. In addition to individual whistle classifications, Rocca will also classify the Detection Number to species. This School Classification is determined by summing the individual classification stats for each species within that Detection Number and selecting the species with the highest total.



1. Detection number: the current detection number. This is the detection number used when a whistle is selected from the spectrogram display.

2. Scroll buttons: scroll through the list of detection numbers.
3. Classification results: the individual classification tally for the current detection number. The list of possible species is based on the classifier model currently loaded. The number beside the species name indicates the number of whistles [classified](#) to that species.
4. School classification: the overall species [classification](#) for the current detection number.
5. Rename Detection: renames the current detection. Any output files previously saved that use the old detection number in the filename will be renamed with the new detection number. Note that the information *in* the [whistle contour stats save file](#) is NOT updated - the user must modify any references to the old detection number manually. Also note that the user will not be allowed to duplicate detection numbers
6. Save Detection: overwrites the current School Stats file (as defined in the [Rocca Parameters window](#)) with the current list of detections and statistics. School statistics are also automatically saved every 5 minutes.
7. New Detection: create a new detection number
8. Whistle Start: lists the time and frequency of the first user-selected point on the spectrogram
9. Whistle End: lists the time and frequency of the second user-selected point on the spectrogram. Note that once the second point is selected, the portion of the spectrogram is captured in a new [popup](#) window.

[Previous: Whistle and School Classification](#)

[Next: Saved Files](#)

Rocca

Saved Files

Rocca saves 3 different files during whistle classification: whistle clip, contour points and contour parameters. Rocca will also save detection stats automatically every 5 minutes, as well as when the [<Save Detection>](#) button is clicked.

If a database module is being used, Rocca will also save the data to two tables: Rocca_Whistle_Stats and Rocca_Detection_Stats.

Whistle Clip

The whistle clip is saved with a wav format to the [output directory](#). The start and end points of the clip are defined by the start and end points originally selected by the user in the spectrogram popup. The channels saved to the clip file are specified in the [Rocca Parameters](#) window.

The filename is defined by the user in the [Rocca Parameters](#) window

Contour Points

The time/frequency pair for each contour point is saved with a CSV format in the [output directory](#). The duty cycle, the energy in a frequency bin around the peak frequency (as defined in the [Rocca Parameters](#) window), and the RMS value of the amplitude are also saved.

The filename is defined by the user in the [Rocca Parameters](#) window

Contour Parameters

The parameters measured from the current contour, as well as the species classification, are saved with a CSV format in the [output directory](#). The information from each classified whistle is appended to the end of the file and the file is never overwritten. Thus, this file will continue to collect classification information every time Pamguard/Rocca is run.

Other saved information for each whistle includes the sound source, the date and time, and the detection number. The end of each row lists the name of the random forest model, the random forest tree votes for each species and a corresponding list of the species names. The species names were added to each row instead of the header line because the header would be created based on information from the very first whistle contour analyzed only. If the classification model is changed for subsequent analysis, the species list may be different and no longer match the header. By including the species list in the row, the user is always able to verify which species were included in the classification algorithm for that particular whistle contour.

The filename is specified in the [Rocca Parameters](#) window.

If a database module is being used, the data will also be saved to the Rocca_Whistle_Stats table.

School Stats

All detection numbers, with the species classification break-down for each, are saved with a CSV format in the [output directory](#). Also included for each detection number is the cumulative random forest tree vote totals for each species, as well as a species list and the overall school classification (based on the species with the highest cumulative tree vote total).

Each time the school stats are saved, either through the auto-save function or by the user pressing the [<Save Detection> button](#), the file is overwritten in order to update any renamed detection numbers. Since a detection number can be renamed but never deleted, no information will be lost when overwriting an old file during a single Pamguard session. HOWEVER, if Pamguard is closed and restarted the file will be **overwritten with blank data and all prior information will be lost**. The file is checked during Pamguard startup, and if it exists the user is given the opportunity to rename it before it is lost, and/or load the existing data back into the system.

Note that when examining a particular detection number, the user should refer to the species list at the end of the row instead of the species listed in the header. The header information is taken from the first detection number listed; if subsequent detection numbers use a different classification model, the species may change and this change would not be reflected in the header.

The filename is specified in the [Rocca Parameters window](#).

If a database module is being used, the data will also be saved to the Rocca_Detection_Stats table.

[Previous: Rocca Sidebar](#)

Appendix B – Abstracts

Abstracts presented at the 161st Meeting of the Acoustical Society of America, 23-27 May, 2011, Seattle, WA.

Seeing the species through the trees: using Random Forest classification trees to identify species-specific whistle types

J.N. Oswald, J.V. Carretta, M. Oswald, S. Rankin, and W.W.L. Au

Acoustic identification of delphinid species is hampered by high variability in whistle characteristics. It is possible that not every whistle contains species-specific information and that there are “species-specific” whistle types. Random forest analysis was used to examine whistles of 8 species recorded in the eastern tropical Pacific Ocean (*Delphinus* species, *Globicephala macrorhynchus*, *Pseudorca crassidens*, *Stenella attenuata*, *S. coeruleoalba*, *S. longirostris*, *Steno bredanensis*, *Tursiops truncatus*). Fifty-one variables were measured from 2,176 whistles. The number of trees within a random forest that “voted” for the predicted species was used as a measure of the strength of classification. A whistle was considered strongly classified if the predicted species received at least 40% of the votes, even if the prediction was incorrect. The percent of whistles that were strongly classified ranged from 33% (*S. longirostris*) to 73% (*G. macrorhynchus*). Overall, 62% of strong whistles were correctly classified, ranging from 22% (*S. longirostris*) to 87% (*P. crassidens*). Overall correct classification for weakly classified whistles was 33% and ranged from 17% (*Delphinus* spp. and *G. macrorhynchus*) to 55% (*S. bredanensis*). Results suggest that while there may be “species-specific” whistle types, the distinctiveness of these whistles and the frequency with which they are produced varies among species.

Integration of ROCCA (Real-time Odontocete Call Classification Algorithm) into PAMGUARD signal processing software

M. Oswald, J.N. Oswald, M.O. Lammers, S. Rankin, and W.W.L. Au

ROCCA is a tool for real-time acoustic species identification of delphinid whistles. Introduced in 2006 as Matlab-based software, ROCCA is currently being incorporated into PAMGUARD, a freely-available, open source software package. ROCCA provides automated extraction of fundamental whistle contours from a spectrogram. It measures 53 whistle contour features, including frequencies, slopes, duration, and variables related to the positions of inflection points and steps. ROCCA currently classifies whistles of seven species and one genus: *Globicephala macrorhynchus*, *Pseudorca crassidens*, *Steno bredanensis*, *Stenella attenuata*, *S. coeruleoalba*, *S. longirostris*, *Tursiops truncatus* and *Delphinus* species. The classifier is a Random Forest trained on 2231 whistles collected over 6 cruises and 7 years in the eastern tropical Pacific Ocean. The original ROCCA classifier used a combination of discriminant function analysis and CART algorithms on 13 whistle contour features for an overall correct classification rate of 35%, which was significantly greater than the random rate of 12%. The current Random Forest scheme, trained on 53 whistle contour features, yields a 62% overall correct classification rate. Feedback from at-sea beta testing has been incorporated into the latest version. Additional species, automated detection, and classification schemes are being explored to extend ROCCA into different geographic areas with greater accuracy.

Abstracts to be presented at the 5th International Workshop on Detection, Classification, Localization and Density Estimation of Marine Mammals, 22-25 August, 2011, Mount Hood, OR

Are there species-specific whistle types that can be used to increase classification success?

Julie N. Oswald, James V. Carretta, Michael Oswald, Shannon Rankin, and Whitlow W.L. Au

Whistles produced by delphinids are highly variable, which makes acoustic identification of these species challenging. It is possible that not every whistle contains species-specific information and that there are “species-specific” whistle types. Random forest analysis was used to examine whistles of one genus and seven species recorded in the eastern tropical Pacific Ocean (*Delphinus* species, *Globicephala macrorhynchus*, *Pseudorca crassidens*, *Stenella attenuata*, *S. coeruleoalba*, *S. longirostris*, *Steno bredanensis*, *Tursiops truncatus*). Fifty-three variables were measured from 2,176 whistles. In random forest analysis, a number of classification trees are created based on measured variables and whistles are classified as the species that the greatest number of trees “votes” for. In this analysis, the number of trees that voted for the predicted species was used as a measure of the strength of classification. A whistle was considered strongly classified if the predicted species received at least 40% of the votes, even if the prediction was incorrect. The percent of whistles that were strongly classified ranged from 33% (*S. longirostris*) to 73% (*G. macrorhynchus*). Overall, 62% of strong whistles were correctly classified, in contrast to a 47% correct classification score for all whistles combined (both strongly and weakly classified whistles). Correct classification scores for strong whistles ranged from 22% (*S. longirostris*) to 87% (*P. crassidens*). Results suggest that while there may be “species-specific” whistle types, the distinctiveness of these whistles and the frequency with which they are produced varies among species. The use of random forest votes as a measure of the strength of classification has been incorporated into ROCCA (Real-time Odontocete Call Classification Algorithm), a Java-based software package for the acoustic identification of tonal sounds that is available as a module in PAMGUARD, an open source signal processing software suite.

Whistle Contour Extraction and Classification using ROCCA (Real-time Odontocete Call Classification Algorithm), a new module in the PAMGUARD signal processing software suite

Michael Oswald, Julie N. Oswald, Marc O. Lammers, Shannon Rankin, and Whitlow W.L. Au

ROCCA is a tool for real-time acoustic species identification of tonal signals. Introduced in 2006 as Matlab-based software, ROCCA has been incorporated into PAMGUARD, a freely-available, open source software package. ROCCA provides semi-automated extraction and classification of whistle contours from a spectrogram. It measures 53 whistle contour features, including frequencies, slopes, duration, and variables related to the positions of inflection points and steps. ROCCA currently classifies whistles of seven tropical delphinid species and one genus: *Globicephala macrorhynchus*, *Pseudorca crassidens*, *Steno bredanensis*, *Stenella attenuata*, *S. coeruleoalba*, *S. longirostris*, *Tursiops truncatus* and *Delphinus* species. The classifier is a Random Forest trained on 2231 whistles collected over 6 cruises and 7 years in the eastern tropical Pacific Ocean. The original ROCCA classifier used a combination of discriminant function analysis and CART algorithms applied to 13 whistle contour features for

an overall correct classification score of 35%, which was significantly greater than expected by chance alone (12%). The current Random Forest scheme, trained on 53 whistle contour features, yields an overall correct classification score of 62%. Group classifications are also provided by compiling individual whistle classification results within an acoustic detection. Feedback from at-sea beta testing during the summer and fall of 2010 has been incorporated into the latest version of ROCCA. Additional species (including *Orcinus orca* and Arctic species such as *Delphinapterus leucas* and *Balaena mysticetus*), automated detection, and alternate individual and group classification schemes are being explored to extend ROCCA into different geographic areas with greater accuracy.