Ishmael 1.0 User's Guide

ISHMAEL: Integrated System for Holistic Multi-channel Acoustic Exploration and Localization

David K. Mellinger[1]

Pacific Marine Environmental Laboratory
7600 Sand Point Way NE
Seattle, WA 98115-6349

[1]Also at: CIMRS, Oregon State University
 Hatfield Marine Science Center
 2115 S.E. OSU Drive
 Newport, OR 97365-5258

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA/OAR. Use of information from this publication concerning proprietary products or the tests of such products for publicity or advertising purposes is not authorized.

Contents

Ishmael 1.0 User's Guide

ISHMAEL: Integrated System for Holistic Multi-channel Acoustic Exploration and Localization

David K. Mellinger[1]

1. Introduction

1.1 What is Ishmael?

Ishmael is a program for acoustic analysis. It contains a spectrogram viewer, three acoustic localization methods, three methods for automatic call detection, real-time sound recording, a beamformer, and a log file annotation feature. It is more or less a collection of methods that have been found useful for analyzing acoustic data sets.

Ishmael's capabilities are primarily aimed at processing large amounts of sound data quickly and relatively easily. The sound can be a collection of sound files, or a signal arriving in real time from one or more microphone(s) or hydrophone(s).

The most basic operation in Ishmael is viewing a spectrogram. The spectrogram is a method for displaying sounds for visual inspection and analysis. A spectrogram shows time on one axis (in Ishmael, the horizontal axis) and frequency on the other axis. If you've never used spectrograms before and wish to learn more about them, I recommend *Principles of Animal Communication* (Bradbury and Vehrencamp, 1998). Another way to learn is to experiment with Ishmael, preferably with someone more knowledgeable nearby to answer occasional questions.

Ishmael runs in Windows. It has been tested, so far, in Windows 95, 98, 2000, and NT. The hardware must be a 486 or Pentium type processor. The amount of memory needed depends on the operating system; use at least 32 Mbytes for Windows 95 and 98, and 64 MBytes for Windows 2000 and NT. The screen size should be at least 800×600 so that all of the dialog boxes fit.

1.2 What Ishmael Isn't

Ishmael is *not* particularly well-suited to sound exploration—taking an unknown sound and examining it in detail to find out and measure its characteristics. For that, I recommend Canary (for the Mac; Charif *et al.*, 1995), Osprey (cross-platform [in Matlab]; available in MobySoft), or probably the

---

[1]Also at: CIMRS, Oregon State University, Hatfield Marine Science Center, 2115 S.E. OSU Drive, Newport, OR 97365-5258

upcoming Raven (cross-platform; from the Cornell Laboratory of Ornithology).

2.   Obtaining and Installing Ishmael

Go to the MobySoft web site, `http://cetus.pmel.noaa.gov/cgi-bin/MobySoft.pl`. In the search box, type in Ishmael, then follow the links to download the latest version of the Ishmael installer. After downloading it, unzip it, then double-click on Setup.exe. Ishmael will be installed in C:\Program Files\Ishmael\Ishmael 1.0 unless you specify some other place. The installer also makes a link so Ishmael will appear on Windows's **Start** button, under Programs. If you like, you can place a shortcut for Ishmael on the desktop: find C:\Program Files\Ishmael\Ishmael 1.0\bin\Ishmael.exe, right-click on it with the mouse, and choose Create Shortcut. Then move the shortcut onto your desktop.

You might have received an Ishmael update as a zip file that has just a collection of .dll files plus Ishmael.exe when you unzip it, and no Setup.exe. You've gotten a *binary* version of Ishmael, which requires that you've already installed a previous version of Ishmael on your target computer. In this case, unzip the .dll and .exe files into the bin directory where you installed Ishmael (e.g., C:\Program Files\Ishmael\Ishmael 1.0\bin).

3.   Getting Started

Start up Ishmael by executing it from **Start→Program Files→Ishmael** at the lower left corner of the screen, or by double-clicking on the Ishmael.exe icon. You'll get a window with some buttons, scrollbars, etc., and a large blank area. If you get warning messages that mention not finding a sound file, you can safely ignore it for the moment.

The next step depends on whether you wish to analyze sound from a sound file or from a real-time sound input source.

3.1   Analyzing a sound file

Ishmael can read and write several common file formats: WAVE (.wav), AIFF (.aif), Sun Audio (.au), and files containing plain integer or floating-point numbers.

To open a file, choose **File→Open file** from the menu. Choose your file and click **Open**. You'll get a dialog box with the file name, a time for Ishmael to start processing sound within the file, and a choice of channels. Set the file start time to 0 and select the channel(s) you wish to see. Click **OK**.

At this point the main Ishmael screen will be visible again. Click on the **Run** button—the one with the green right-arrow—to start the sound analysis and display process. See "Analysis and display options" below for further options.

3.2 Analyzing real-time sound

Ishmael currently has interfaces for three real-time sound input devices:

- Stereo input from a sound card, any Windows-standard sound card. If you have a sound card with more than two channels, it should work for multi-channel input in Ishmael too.

- Multi-channel input from National Instruments E-series boards. The ones that have been tested are the PCI-6071E internal board, with up to 32 channels, and the DAQCard-6062E (PCMCIA) card, with up to 16 channels.

- Multi-channel input from Data Translation 3800 series boards (3801, 3808, 3809, 3814, or 3818). The only one that has been tested is the 3809.

After you install one of these cards in your computer (including installing the software interface), a menu item for it appears in Ishmael's **File** menu. Choose this menu item, and a dialog box appears. Choose the channels you want to input, the sampling rate, and any other options that are relevant. If you have more than one board in the computer, you'll have to choose which one you want. Click **OK**.

At this point the main Ishmael screen is visible again. Click on the **Run** button—the one with ▶, the green right arrow—to start the sound analysis and display process.

4. Analysis and Display Options

Once you click the **Run** button (the green right-arrow in the main Ishmael window), you should see sound data scrolling across the screen. There are various ways to control what you see.

Note: If you are analyzing a sound file, the scrolling may finish quickly, in which case changing the display options listed here will have no immediate effect. After changing one of the options below, click the **Run** button again to re-display the sound file.

Note: Some of the options listed below can't be done while Ishmael is analyzing sound. If the control (button or menu choice) you want is grayed out, you need to click the **Stop** button first—the red square—to activate the control.

4.1 Signal waveform and spectrogram

Ishmael currently has two ways to display sounds, the signal waveform (or time series) and the spectrogram. You may display both at once if you like. To turn these on or off, use the **View→Signal waveform** and **View→Spectrogram** menu choices, respectively.

4.2   Brightness and contrast

If what you see is all black or all white, it's probably because the brightness and/or contrast are adjusted badly. You can fix these by changing the scroll bars in the upper right corner of the window. The top scroll bar is brightness; slide it along until you start to see something interesting in the main display. You can similarly change the contrast. Typically, you want to set brightness and contrast so that some background noise is just barely visible, and any animal sounds show up strongly against the background noise.

4.3   Time and frequency scaling

The sound you are analyzing may zip across the screen too fast to see, or conversely may crawl along too slowly to see much detail. To fix these, choose **View→Time scaling** from the menu. Type in a new numeric value for the time scaling and click **OK**. Click the **Run** button to restart processing.

Similarly, Ishmael may be showing the wrong frequency band of the sound you're interested in. One symptom of this is having your spectrogram occupy only a small part of the display band, with a black region either above or below the data. On the menu, choose **View→Frequency range...** and enter an appropriate value. The frequencies available range from 0 to one-half of the sampling rate, so if your sound input (in a sound file, or from real-time input) is set at, say, 22 kHz, then the display will show data between 0 and 11 kHz.

4.4   Spectrogram parameters

The spectrogram analysis parameters determine the time and frequency resolution of the spectrogram. These control how much detail you can see in your sound in both time and frequency. These resolutions are inversely related, so that choosing finer time detail will give you coarser frequency detail, and vice versa. There is not the space here to describe how to choose appropriate values for these parameters, though you are welcome to experiment with them and see what you get; Ishmael won't crash. The book mentioned above, *Principles of Animal Communication*, is a good place to learn more. To set the analysis parameters for your spectrogram, choose **Compute→Spectrogram parameters** from the menu. If you're not sure where to start, try these values:

> **Basics**
>> Frame size   256 (see below)
>> Zero padding   None
>> Hop size $\frac{1}{2}$
>
> **Advanced**
>> Window type   Hamming
>> Intensity scaling   logarithmic
>> (the choice buttons) the same duration in seconds

For the frame size, start with 256 samples. Click **OK**, then click the **Run** button to start the spectrogram analysis and display. Change the brightness and contrast scrollbars (the ones at upper right) to make your spectrogram data clear. Experiment with large and smaller values of the frame size, and see what detail is visible in the resulting spectrogram. Larger frame sizes give you better frequency resolution, while smaller ones give you better time resolution.

Technical note: Ishmael computes a spectrogram by taking "Frame size" samples of the input signal, multiplying by the window (Hamming or whichever), appending "Zero padding" samples of zeroes, and calculating the Discrete Fourier Transform of the result. It then takes the bins corresponding to positive frequencies, scales them by the "Intensity scaling" method, and uses the result to paint one vertical stripe in the spectrogram. Then it advances by "Hop size" × "Frame size" samples and repeats the process.

Note: Other spectrogram programs have other ways of describing spectrogram parameters. These are related to Ishmael's parameters as follows: "FFT Size" is the sum of Ishmael's Frame Size and Zero padding. "Overlap" is the complement of Ishmael's Hop size, or [1 – Hop size]. Overlap is sometimes specified in samples, in which case it is [(1 – Hop size) × Frame size], or in seconds, in which case it is [(1 – Hop size) × Frame size / Sample rate].

4.5   Equalization

Equalization can be very useful for viewing animal sounds. Equalization, a form of automatic gain control, averages out the absolute level of a spectrogram, so that the background noise level stays roughly constant over time. It operates on each frequency band—each horizontal stripe—in a spectrogram. For this reason, it helps eliminate continuous narrow-band sounds, such as electrical hum, motors, and so on. It also helps *whiten* a spectrogram, making all frequencies equally intense over the long run.

Choose **Compute→Equalize 'gram** to enable equalization. Equalization has a time constant associated with it that controls the amount of time it takes for the equalized spectrogram to reach the background level. If this time constant is too short, equalization will "equalize away" the animal calls you are interested in. That is, equalization may happen so fast that after the call starts, the equalizer reduces the spectrogram level in the frequency band of the call, making the later part of the call fade into the background spectrogram level. If the equalization time constant is too long, then it takes the equalizer too long to start up—to take effect after you click the **Run** button—and to recover from loud sounds. A good place to start is to make the equalization time constant 2–3 times as long as your call type of interest. Click the **Run** button, see the resulting spectrogram, and adjust the time constant longer or shorter as appropriate.

5.   Storing Settings

Ishmael can save the values you have specified for various settings—the name of the sound file to analyze, the real-time input card parameters, the spectrogram calculation parameters, and so on. To save these settings, just choose **File→Save settings as** from the menu. The file you save the settings in is called a *settings file.* Later, you can choose **File→Load settings** to retrieve the values you saved.

Ishmael loads a certain settings file when it first starts up. This file is C:\Program Files\Ishmael\Ishmael 1.0\IshDefault.ipf (or the same filename in whatever directory you installed Ishmael in). You can save your current Ishmael settings in this file by choosing **File→Save settings as default**. Then the next time you start Ishmael, these settings will be loaded and used.

Settings files are plain text files. You can copy them, store them in the same directory as your sound files, and generally treat them as you would any other file. You can also edit them in a word processor if you like; be sure to save a settings file as plain text, not in the format of your word processor. The names of most parameters in the settings file correspond fairly well to the names that are in Ishmael's various dialog boxes. One very useful thing you can do is to make a settings file that has only a few chosen parameters, like the time scaling you like, or the automatic detection parameters you need. When you load a settings file like this, only the values you specify are changed; Ishmael keeps its current values of all the parameters that aren't explicitly changed by the settings file.

6.   Batch Processing

You can view a large batch of files easily, without having to pick each one out in the file-opening dialog box. To do this, simply choose **File→Open file** from the menu, then select two or more files. Use Shift-click or Ctrl-Left Click (i.e., hold down the keyboard's Shift or Control key while you click with the mouse) to add more files to the current set. Or you can type **Ctrl+A** (i.e., hold down Control and press the A key) to select all the files in a directory. After you click **OK** and get back to the main Ishmael window, click the **Run** button as usual to view the spectrogram of the first sound file in the list. To go to the next file, choose the menu item **Run→Next batch file**. The current file name is displayed in the title bar of Ishmael's window.

7.   Recording

Ishmael can record sound as it acquires it, via the **Record→Record sound** menu item. The box that appears lets you specify where you want recorded sounds stored—that is, the directory in which to save them. The sound file names themselves are set from the **File→Saved file names** menu item.

You can also specify the length of each sound file. Sound data accumulates rapidly, and sound files can easily become unmanageably large. Specify a maximum length for your sound files as hours:minutes:seconds. The hours

and minutes fields are optional, so "15:00" means 0:15:00, and "200" means 200 seconds, the same as 0:03:20. Ishmael will start a new sound file when the file it is recording reaches this length. If you don't want a maximum length—if you always want to record into a single sound file—then specify a very large value for the maximum length, like 100:00:00.

The second tab, **Sampling**, allows you to turn recording on and off on a pre-set schedule. The options here are fairly straightforward.

WARNING: The standard sound card interface in Windows provides no way for Ishmael to tell if sound samples have been lost, that is, if something in Windows delayed the processor and made the sound card interface omit some samples. If you intend to use Ishmael for scientific recording, it is recommended to get one of the data acquisition boards from Data Translation or National Instruments mentioned in "Getting started" above. Those boards' interfaces do allow Ishmael to tell if samples are lost.

8.   Logging Comments

A *log file* is a plain text file to which Ishmael appends text upon request. Open a log file using the **Actions→Open log file** menu item. Pick a file name and click **OK**.

To add text to the log file, just choose **Actions→Log a comment** from the menu. Type in your message, then click OK or press **Ctrl+Enter** to write the message to the log file. Log file comments are always appended to an existing file. To clear the log, choose **Actions→Clear log file** from the menu.

Logging can also happen as a consequence of Actions and by means of automatic detection. See below for details.

9.   Localization

When set up correctly, Ishmael can do localization—determining the location of a sound source—in either one dimension, calculating a bearing angle, or two dimensions, calculating an X-Y position. Having Ishmael set up correctly for localization means it needs these things:

1. Sounds with two or more channels, either from a sound file or from real-time input. Ishmael's methods for determining bearings require at least two channels of sound (two simultaneously operating microphones or hydrophones), and the X-Y position method requires at least three channels.

2. Information about the positions of the microphones or hydrophones.

3. Information about the speed of sound.

9.1   Configuration

To set up Ishmael for doing localizations, first create a text file with the positions of your microphones/hydrophones. This file, the *phone array file*,

should have one line for each phone, specifying two values per phone: the X- and Y-positions of the phone, in meters, with spaces or tabs between the two numbers in the file. The positions should be relative to some origin or (0,0) point. This origin can be anywhere you like, typically at one of the phones or (for towed hydrophone arrays) at the ship. An example phone array file for a four-phone array looks like this:

```
-100    20
-20     150
100     30
75      -30
```

Save your file, being sure to save it as "text only" in your word processor. In Ishmael, choose **Localize→Loc options...** from the menu, choose the **Basic info** tab, and enter the name of your phone array file.

Also enter the speed of sound here—in the ocean, it's typically around 1500 m/s; in air, typically around 340 m/s.

What you do next depends on the type of localizations you want to do. In the dialog box, pick the appropriate tab:

1. **Phone pair** bearing calculation. A pair of phones determines a hyperbola on which the sound source lies. When the phones are close together, this hyperbola is very nearly a pair of bearing angles, one on the left side and one on the right side of the line joining the two phones. When you localize a sound using this method, bearing angles are plotted on a chart that shows only one of the two bearing angles, the one between 0° and 180°. Choose which two phones you wish to use for the bearing calculation.

2. **Beamforming** bearing calculation. This is a method of calculating bearings that relies on a frequency beamforming algorithm developed by Aaron Thode. It works well with many hydrophones, though you can use it with as few as two. It's useful when the sound source is relatively far from the hydrophone array—say, several times the separation of the most distant phones. Currently, this method assumes that the phones lie in a straight line, and that the bearings therefore have left-right ambiguity, similar to phone-pair bearings. There are two different variants of the beamforming bearing calculation, one that works well for pulses, clicks, and other impulsive sounds, and one that works well for whistles, moans, and other tonal sounds. Choose the appropriate method for the type of call you'll be trying to locate. For tonal sounds, also specify the instantaneous bandwidth of the frequency contour— that is, specify how tall the contour is in a spectrogram, in Hertz. Results are plotted as for the phone pair bearing method above.

3. **Hyperbolic** X-Y position calculation. These locations are X-Y positions calculated by using differences in the times that sounds arrive at multiple phones. Conceptually, the time difference for each pair of phones determines a hyperbola on which the calling animal lies. The

intersection of these hyperbolae determines the animal's X-Y position. This localization method is useful when the sound source is relatively close to the phone array—say, closer than several times the separation of the most distant phones. Positions are plotted on a map; you determine how big the map is using the four values specified in this dialog box. Choose values large enough to encompass both your phone array and some space around it.

Also, you need to be sure that Ishmael is getting enough channels of sound—at least two or three channels, depending on the localization method used. If you are analyzing a file, choose **File→Open file...** from the menu; if you have already selected a file, just cancel the file-picker dialog box, and Ishmael will leave the same file open and let you choose which channels you want to analyze. Select two, three, or more. If you are analyzing sound in real time from a sound card or other sound input board, make sure at least two or three channels are input.

For any localization method, you will probably want to see all available channels on the screen. (Ishmael allows you to have input channels that it uses for localization but does not display, but usually it's better to see all the channels Ishmael is using.) On Ishmael's main menu, choose **View→All channels**.

9.2   Localizing sounds—most methods

Run Ishmael until you see an animal sound you wish to localize. Sounds that span a wide range of frequencies tend to work best for localization, as they resolve ambiguities better than narrow-band sounds. If you wish, pause the processing by pressing the space bar or clicking the **Pause** button (the ⏸ one with the two black bars), or halt the processing by clicking **Stop** button (the ⏹ one with the red square).

Select the sound to locate with the mouse by drag-selecting a box around the sound. If you do this drag-selection without first halting the spectrogram, Ishmael will automatically pause the spectrogram. The more tightly you can draw this box around the sound, the less noise is included in the localization calculation, and the more accurate the location will be. However, you should be sure that the box you draw encompasses your animal sound in all of the channels; since the sound on some channels may be slightly delayed relative to others, you may need to draw the box a bit longer on the left or right.

Once you've drawn the box, choose from the **Localize** menu: **Beamforming bearing, Phone pair bearing**, or **Hyperbolic loc**.

9.3   Localizing sounds—beamforming bearings with tonal calls

Instead of drawing a box around the call you wish to localize, you must highlight the frequency contour by tracing over it. Hold down the Control key and left-click with the mouse at the start of the contour. Move a bit farther along the contour and do another Ctrl-Left Click. You'll see a straight line joining your clicks. Move a bit farther and do another, and the line will

grow longer. Keep going until you reach the end of the contour, then choose **Localize→Beamforming bearing** from the menu.

9.4   Viewing localization results—all methods

Localization methods that produce bearings—both phone pair and beamforming methods—display their results as an angle between 0° and 180°. Each bearing shows up as a blue dot on the display, and the angle of the most recent bearing is also shown in the title bar of the display window. Bearings are spread out in time in this window, so you can see how the bearings change over time. To expand or compress the time scale in this display, click the yellow-and-black buttons at the top of the window.

If you are using Ishmael in conjunction with WhalTrak or another standalone plotting program (a topic beyond the scope of this User's Guide), you can click the **Send To WhalTrak** button. This will send the current bearing to the serial port as a text string, and the bearing will turn red on Ishmael's bearing display. The serial port is configured to use 9600 baud, no parity, one stop bit.

The localization method that produces an X-Y position—the hyperbolic method—plots its results on an X-Y diagram, called the *map*. Hydrophone positions are shown in red on the map, and the calculated positions are shown in white.

Sometimes the localization algorithm fails to come up with a valid bearing or position. In this case, the window's title bar will tell you this; no new bearing or position is plotted.

For any of the localization methods, you can clear the display by choosing **Localize→Clear map** from Ishmael's menu.

9.5   "Instant" localizations

Ishmael has a feature for doing localizations quickly, called Insta-Loc. It's most useful when analyzing a real-time signal for calls that don't vary in duration or frequency band very much.

Here is how it works: You set parameters that specify the duration and frequency band of a call. When you click in the spectrogram while holding down Shift and Control (Shift+Ctrl+LeftClick), Ishmael automatically pauses the scrolling and makes a selection around the point where you clicked. Then it calculates a localization, displays the result (as a bearing or X-Y position, whichever is appropriate), and un-pauses the scrolling. So with a single mouse click, you get a localization.

10.   Beamforming

Beamforming is a way of combining sounds from two or more microphones or hydrophones that allows you to preferentially hear sounds coming from certain directions. The method used in Ishmael is a delay-and-sum beamformer, in which sounds from each phone are delayed relative to sounds from the other phones, and the delayed signals are added. The amount of

delay determines the *beam angle*—the angle to which Ishmael preferentially "listens." When an animal call arrives from this angle, the sound signals from the multiple phones are added constructively. The resulting sum is stronger, and the call shows up in Ishmael relatively well. When a sound arrives from another angle, the delayed signals from the various phones add destructively—with positive and negative parts of the sound waves canceling out to some degree—and the sum is not as loud as an equivalent sound arriving from the beam angle.

The amount of gain you get from beamforming depends on several factors, most importantly on the number of phones. You get some improvement with only a handful of phones, but it really takes more than five or six phones to get much gain, and eight is recommended. Many commercial applications of beamforming use 20, 50, or more phones. Another important factor is *coherence*. Coherence is a complex topic, but its impact is that beamforming performance can be degraded significantly if your phones are too far apart. As a very rough rule, keep microphones (in air) within a few meters to tens of meters, and hydrophones (in water) within tens of meters to perhaps 2–300 meters. More variable or turbulent environments require closer placement of phones, and more homogeneous environments permit wider spacing. Beamforming performance also depends on the frequency of the signal: higher frequencies result in narrower beams, but they are subject to *sidelobes*, angles other than the intended one that have strong gain. Lower frequencies, conversely, have broad beams and fewer sidelobes. The overall response of a beamformer is usually expressed as a *beam pattern*, a circular plot showing the amount of gain for each angle from which sound can arrive.

To use beamforming in Ishmael, choose **Compute→Beamforming** from the menu. Specify the beam angles you wish to use, and check the appropriate choice in the x-axis/y-axis box. It is recommended to check the **Enable weighting** box. Weighting here is comparable to windowing in computing a spectrogram, and produces a beam pattern with fewer sidelobes. Ishmael uses Hamming weighting.

If you have installed Matlab on your computer, you can view the beam pattern you will get. Click on the **Plotting** tab and follow the instructions to plot a beam pattern, which shows the beam response in decibels as a function of angle. Note that the beam pattern is dependent on some factors that do not appear on this tab: The sampling rate of the input signal, the positions of your phones (which are set via **Localize→Loc options)**, the **Enable weighting** check box, and the x-axis/y-axis choice all affect the beam pattern. (Note: In Matlab version 5 and earlier, the option of making polar plots with zero degrees on the y-axis is not available, so beam patterns are always shown with zero degrees on the x-axis. Regardless of this, beams in Ishmael really do have zero degrees on the y-axis, if that is what you specify.)

11.   Automatic Detection

To use automatic detection in Ishmael, you need to specify a number of settings and parameters. Here is a brief guide.

11.1   Choose a method

Ishmael currently has three basic methods for automatic detection, and many options that affect the operation of the detection methods. The first thing to do is to choose a method:

- **Energy summation.** In this method, the values in each vertical strip of the spectrogram are summed. The sequence of sums, one sum per vertical strip, makes up the detection function. Energy summation is a useful detection method for scanning sounds to find everything of interest, particularly when the target sounds may be unknown or highly variable. It can also be used as a first step for a more sophisticated multi-step classification system.

  The summation is done between two frequency limits that you set via the **Detect>Energy sum** menu item. Actually the result is the average, not just the sum, of the elements along each vertical strip, so if you change the frequency limits of the operation, the detection function result stays about the same when only background noise is present. Typically you set the frequency limits of the sum to include the frequency range of your call type of interest.

  A variant of energy summation uses the ratio between the energy in two frequency bands as the detection function. In this case, the energy averages in each of the two frequency bands that you specify are computed along each vertical strip. The ratio between these energy averages gives the detection function.

- **Matched filtering.** This method works by cross-correlating Ishmael's input signal with another signal that you specify, called the *kernel*. Cross-correlation is the optimum method for detecting a known signal in white Gaussian noise (Van Trees, 1968). It works best when the signal you wish to detect is quite constant from one instance to the next and from one animal to the next.

  Matched filtering is typically done with either a natural or a synthetic kernel. To use a natural signal, find the clearest, most typical example you can of the call you wish to search for. This can be done using Ishmael's spectrogram display capability or any other method you like. Select the call, making the selection just long enough to contain the call and nothing more. Choose **File→Save selection as** and save the call as a sound file.

  Constructing synthetic kernels of animal calls (e.g., Dooling *et al.*, 1982; Buck *et al.*, 2000) is beyond the scope of this discussion. If you do it, construct your synthetic kernel at the same sampling rate as the sound signals you wish to analyze, then save it as a sound file.

For either a natural or synthetic kernel, choose **Detect→Matched filter** and enter the name of your kernel sound file.

- **Spectrogram correlation.** This method works by cross-correlating a spectrogram of Ishmael's input signal with a synthetic time-frequency kernel. It works well when there is some amount of variability in the call type to be detected; certain parameters in construction of the kernel can be arranged to allow more or less variability in the calls to be detected. The construction of kernels, and their performance, is beyond the scope of this guide, but more information is available elsewhere (Mellinger and Clark, 2000).

  To use spectrogram correlation, choose **Detect→Spectrogram correlation** from the menu. Enter the contour width, which will be applied to all segments of the correlation kernel, and then click the **Define contour** tab. Enter the time/frequency endpoints of each segment of the kernel.

11.2   Condition the spectrogram

The two detection methods that rely on spectrograms, energy summation and spectrogram correlation, are affected by how Ishmael calculates the spectrogram and also how it conditions the spectrogram—whether equalization, spectrogram floor, and spectrogram ceiling are enabled. Equalization is recommended for animal call detection, as it helps eliminate some common types of noise like motors and 60 Hz electrical noise. See the "Analysis and display options" section above for information about equalization.

The choices in **Compute→Set floor in 'gram** and the **Compute→Set ceiling in 'gram** affect the spectrogram-based detection methods too. The floor and ceiling enforce minimum and maximum values on the spectrogram, respectively. This is useful when, as often happens, there are occasional very low and very high values in the spectrogram that overwhelm the detection methods. Very low spectrogram values can happen simply from random background noise, very high values from loud calls or interfering sounds. To limit the impact that these have on automatic detection, enable the floor and ceiling. When **automatic** is checked, the floor or ceiling value is calculated from the setting of brightness and contrast when you start a Run. The floor value is the level at which sound is just visible in the displayed spectrogram, below which all sounds are displayed equally dark. Likewise, the ceiling value is the maximum displayable value, above which all sounds are displayed equally bright.

11.3   Examine the detection function

Each detection method analyzes the input signal and produces a detection function, which is a signal specifying over time the likelihood that the call of interest is present. Run Ishmael with your detection method enabled. A window will appear showing the detection function. You'll almost certainly need to adjust the amplitude in this window using **View→Amplitude range**—in fact, often the first time you run your detector, all that appears is a black

box, because the amplitude limits in this box are adjusted badly and the detection function is outside the limits. The scaling of the detection function varies depending on the type of detection and the various options you choose; it's not closely related to any ordinary unit of measure. The only thing that's really important in it is the height of the detection function relative to your threshold. The display is labeled "U" to mean arbitrary Units.

If you can't see anything at all in the detection function window, try using amplitude limits of 0 ± 100 and clicking the **Run** button again. When you can see your detection function, adjust the amplitude limits again to zoom in until you can see the detection function varying with time. You should see peaks in the detection function where your calls of interest occur, and hopefully not in many other places.

Note: If you ever change the width of the main Ishmael window, the width of the detection function window does not change, and its time axis no longer lines up with the main window axis. To fix this, just close the detection function window; the next time you start a Run, the detection function window will be re-created at the right size.

11.4   Smooth the detection function

The detection function is often very jumpy over time. It's usually a good idea to even it out using smoothing, available via the **Detect→Detection options** menu choice. As a rule of thumb, smoothing time constants that are roughly as long as your call type of interest seem to work well. But you can play with this option a bit to see what works best: What you are trying to achieve is a smoothing time constant that produces peaks in the detection function where your calls of interest occur and evens out any other peaks due to background noise. Beware that changing the smoothing constant will change the absolute level of the detection function; what is important is the relative height of peaks corresponding to calls and peaks corresponding to background noise. You want "call peaks" that are high relative to "noise peaks."

11.5   Set a threshold

You need to set a threshold in this detection function; whenever the detection function exceeds the threshold, a detection event is triggered. Examine the detection function and pick a value on the y-axis that is less than the height of the peaks in the detection function but greater than the normal, background level of this function. In the **Detect→Detection options** menu choice, on the **Threshold** tab, enter the threshold value you have picked.

Actually, the detection function has to be over the threshold for a certain amount of time to trigger a detection event. Set the time limits, both the minimum time and maximum time over threshold, using this same **Threshold** tab.

11.6   Set a neighborhood

Also on the **Threshold** tab is the **Detection neighborhood** time. This value prevents detections from happening too often, and it's useful for at least two reasons. First, it prevents echoes and other multiple-arrival effects from triggering multiple detection events. Second, the detection function often has a bit of jitter to it—fast, fine-scale movement up and down. As the function crosses the detection threshold, this jitter can trigger a spurious detection. Setting a detection neighborhood can prevent this. A good rule of thumb is to use a detection neighborhood value that is about half the time between the end of one call and the start of the next.

The neighborhood specifies the minimum time after the end of one call—after the time the detection function goes back below the threshold—before another detection event can be triggered. If this neighborhood time ends while the detection function is already above threshold, a new event is not triggered.

11.7   Set up regular sequence detection

Ishmael has a method of analyzing the detection function to look for regular sequences. It works well for very regular sequences; if there is even a bit of irregularity in the timing of calls, it does not work very well. If your species produces calls at very regular intervals, try this and see whether it improves detection performance.

Choose **Detect→Regular sequences** from the menu to enable it. When enabled, Ishmael does the following calculation repeatedly (Mellinger and Clark, 1997): It takes a portion of the detection function that is "window length" seconds in duration, calculates the autocorrelation, finds the largest value occurring between the **min** and **max** repetition periods, and outputs that value. Then it moves forward by "window hop size" seconds and repeats the operation.

11.8   Specify what to do with detection events

In the **Edit actions→Call detected** box, or equivalently under the **Saving calls** tab of the **Detect→Detection options** menu choice, specify what you want to do when a call is detected. There are many options here; see the "Actions" section of this guide for details. One of the options is "save the current selection." In the context of automatic detection, the "current selection" is the call being detected. More precisely, it's the time span from when the detection function went above threshold to when it went back below, plus any extra time before and after that you specify on the **Saving calls** tab.

Also, in the **Edit actions→Call detected** box, there are options that make no sense in the context of automatic detection: "continue running (unpause)" and "go to the next file in a batch run." These options are meaningless since Ishmael isn't paused when it detects a call. No promises are made about what will happen if you enable these options here.

11.9 Specify file names of saved calls

If you have enabled the saving of detected calls, you need to specify their names via the **File→Saved file names** menu choice. File names can include the time and date that the call occurs; this really works only for real-time input, since Ishmael currently can't extract timestamps from most types of sound files. Also specify any amount of extra sound to save before and after the call occurs, using the **Saving calls** tab of the **Detect→Detection options** menu item. If you are saving detected calls in sound files, it is strongly recommended that you do a trial run and look at a few of these saved files to make sure that enough time gets saved before and after each call. Sometimes the automatic detector's idea of when a call starts is different from your idea.

11.10 Specify the log file

If you have enabled any of the log file options in the **Edit actions→Call detected** box, you need to open a log file. Do this with the **Actions→Open log file** menu choice.

11.11 Run it

That's it! Click the **Run** button and Ishmael will process your sound(s), extracting and/or logging the calls it detects.

12. Automatic Detection Summary

This is a list of the settings in Ishmael that can affect automatic detection. If you wish to get the same result out of successive runs of Ishmael, keep these settings the same. It is strongly recommend to save a settings file that has your automatic detection settings.

1. The input signal.

2. The detection method—energy sum, matched filter, or spectrogram correlation.

3. Everything in the box that appears when you choose **Detect→Detection options**.

4. Everything in the box that appears when you choose **Actions→Edit actions** and then double-click on the **Call detected** item.

5. The name specified in **File→Saved file names** box, if you have checked **Save the selection** in the **Call detected** item. Be aware that the computer's current time, time zone, and daylight-saving time setting can affect the file name, because the **File→Saved file names** box allows you to include a timestamp in your file name. See Windows's **Start button→Settings→Control panels→Date/Time** to change the time settings.

6. The log file settings under **Actions→Open log file**, if you have checked any of the logging options in the **Call detected** action.

7. Beamforming—whether **Compute→Beamforming** is enabled, and if so, its settings.

8. Sequence detection—whether **Detect→Regular sequences** is enabled, and if so, what its settings are.

In addition, if you use a detection method that relies on spectrograms (energy sum or spectrogram correlation), these other parameters matter:

9. Spectrogram parameters. The one that matters the most is logarithmic vs. quadratic scaling. Window type matters somewhat too. The other parameters have correction factors, so changing them should make only minor differences in the detection function.

10. Equalization—whether it's enabled, and if so, its time constant.

11. Floor/ceiling values, if used.

12. *Brightness and contrast of the display*, if floor or ceiling values are used and "automatic" is checked for them. This one is especially easy to miss, since you often change brightness and contrast using the scrollbars and don't really think about it affecting automatic detection. But the automatic floor and ceiling values are calculated from the current brightness and contrast values when you click the **Run** button, and they definitely affect automatic detection. To stop the brightness and contrast scrollbars from affecting automatic detection, check "manual" instead of "automatic."

Finally, here is the order in which Ishmael's sound processing happens. Knowing this order can help in understanding what is affecting automatic detection:

Sound input (from a file or one of the real-time sources)
Sound recording
Beamforming
Matched filtering, if enabled
Signal waveform (time series) display
Spectrogram calculation
Spectrogram equalization
Spectrogram floor and ceiling
Spectrogram display
Energy summation or Spectrogram correlation, if either one is enabled
Detection function sequence analysis
Detection function smoothing
Detection event analysis (applying thresholds, neighborhoods, etc.)
Detection actions—logging, saving a file, etc. (see below)
Detection function display

13.   Actions

An *action* is something you specify that Ishmael performs when you tell it to. You tell Ishmael to perform an action by choosing it from the **Actions** menu, or by typing the action's *hot key*. The things Ishmael can do when performing an action include writing to the log file, saving the current selection as a sound file, and localizing the current selection. If your action includes several of these things, then Ishmael will do them all when you ask it to perform the action.

To create an action, choose **Actions→Edit actions** from the menu, then click **New**. Give your action a name and, if you like, a hot key. The action will be performed when you type the hot key. Beware that hot keys are not checked to see if they are already in use elsewhere in Ishmael. Actions can include the following steps:

- **Continue running (unpause)**. Often you want to execute an action while Ishmael is paused; this action makes Ishmael unpause itself and start running again.

- **Save the selection as a sound file**. If you have selected some sound in the main Ishmael window, this step will automatically save the selection in the directory you specify. The file name of the saved sound is specified in the **File→Saved file names** menu item.

- **Go to next file in a batch run**. If you are processing a large set of files as a batch, this terminates processing of the current file and goes on to the next one in the list.

- **Execute the current Matlab command**. This is the same as choosing **Compute→Do Matlab command** from the menu. If there is a selection, it is sent to Matlab as well.

- **Calculate location of the selection**. This executes the phone-pair localization method. (The other localization methods should soon be available).

- **Logging** (on the **Logging** tab). Various types of information are automatically written to the log file.

The "Call detected" action is special. This action is performed automatically whenever a detection event is triggered (see the "Automatic detection" section above). When this action is performed automatically this way, the current selection will be the detected call; it can be saved to a file or logged, as you specify. This action is otherwise like any other action: you can give it a hot key, or perform it by choosing it from the menu.

14.   Matlab Interface

Matlab is a language for processing and displaying data, with particular emphasis on processing matrices. (The name comes from "matrix laboratory.")

Ishmael has an interface that allows you to send sounds to Matlab and execute Matlab commands. This interface has been tested on Matlab versions 5 and 6. To use it, first choose **Compute→Configure Matlab command** from Ishmael's menu. Type in the command to be executed in Matlab.

You should also specify the directory for Matlab to run in—the launch directory. Ishmael tells Matlab to go to this directory when it first launches Matlab, but not any time thereafter, so if you change directories in Matlab, Matlab will stay in that directory. You have the option of executing the Matlab script named `startup.m` when Matlab is launched.

It is strongly recommended that you read the tab labeled **Matlab tips**.

To execute your command from Ishmael, choose **Do Matlab command** from the menu, or type **Alt+M**. Ishmael sends several types of data to Matlab, then executes your command. The data include the sampling rate, the start- and end-time and frequency band of the current selection, the sound samples of the current selection, the speed of sound and phone positions (from **Localize→Loc options**), and several other items. To see them all, go to the Matlab command window, which will be on the task bar at the very bottom of the screen, and type the command `whos`.

15.   Configurations for Typical Tasks

These are some common tasks done in Ishmael, and the parameters you need to set to make them happen. These are very terse descriptions, and are meant for use after you are somewhat familiar with Ishmael.

15.1   View a sound file to see what's in it

First, disable Ishmael's more exotic options by looking through the menus for check-marks—the only ones you want are **View→Signal waveform** and/or **View→Spectrogram**, and possibly **View→All channels**. Look through Ishmael's remaining menus to see if any other options have check marks (**Compute→Equalize**, **Compute→Set floor**, **Compute→Set ceiling**, **Compute→Beamforming**, **Detect→(anything)**, or **Record→Record sound**), and if they do, disable them.

Open your sound file with **File→Open file**. After clicking **Open**, check **Fit file contents to screen** and click **OK**. On the **View** menu, choose **All channels**, **Signal waveform**, and/or **Spectrogram** if any of these don't have check marks. Click the **Run** button, then adjust the brightness and contrast scrollbars in the upper right corner of the window until you can see data in the spectrogram. Adjust spectrogram parameters to clarify the image using **Compute→Spectrogram parameters**. Adjust **View→Amplitude range** until you can see the waveform clearly. Do **File→Save settings as** if you so desire.

15.2   Convert sound file formats

To convert a sound file from one format to another, first repeat the steps in "View a sound file to see what's in it" above that disable Ishmael's more

exotic options. Choose **Record→Record sound** from the menu and enable recording, then choose the directory to store the converted files in. Set the maximum length of each file to a time longer than any of your sound files—say, 1000:00:00. Un-check "**Time align the sound files**" and "**Record only when getting real-time input**," then click **OK**. Next, choose **File→Saved file names** from the main menu. For the file template, enter   **%f.***exten* where ***exten*** is the extension of the file type you are converting to. For instance, if you are converting WAVE files (with extension .wav) files to AIFF files (with extension .aif), enter **%f.aif** for the file template. The time zone choice here doesn't matter. Click **OK**. Lastly, open your original file using **File→Open file**, then click **OK** and click the **Run** button. Your converted sound file should appear in the directory you chose for recorded sounds.

To convert a number of files en masse, do **File→Open file** and simply choose the files at once using the Shift and Control keys when you click with the mouse, or type **Ctrl+A** to select all of the files in a directory. Then instead of clicking the **Run** button, choose **Run→Batch run** from the menu. Un-check the box labeled **Pause after each file** and start the batch run.

15.3   Automatically find calls in a large archive

Choose **File→Open file**, then select as many of your archive's sound files as you like. (Note: Some earlier versions of Windows can't handle huge collections of files, like more than a thousand or two at a time.) Click **Open**, check **Fit file contents to window** if you like, and click **OK**. Choose **Compute→Equalize 'gram**, enable equalization, and choose a time constant appropriate for the calls you're interested in. Adjust the spectrogram parameters as needed using **Compute→Spectrogram parameters**. Click the **Run** button, then adjust brightness and contrast until the background noise level is just barely visible, or even just less than visible, in the spectrogram. Choose **Compute→Set floor in 'gram** and click both **Use floor value** and **Use ceiling value**, choosing **automatic** for both of them. Choose **Detect→Energy sum**, enable it, and pick appropriate frequency limits. Click the **Run** button again and examine the resulting detection function. Choose **Detect→Detection options** and set appropriate values for smoothing, detection threshold, min/max durations, and detection neighborhood. Under the **Saving calls** tab, set the before/after times, then click the **Edit the action** button. Check the **Save the selection as a new sound file** option, specify a directory there, and uncheck all the other options. Click **OK** until you're back at the main Ishmael window. Click the **Run** button again and see what detections you get; each one will result in a short sound file in the directory you specified for saved files. If necessary, adjust the threshold, smoothing value, etc., until you're happy with the results. Do **File→Save settings** if you so desire. Then choose **Run→Batch run** from the menu, make sure **Pause after each file** is un-checked, and click **OK** to start the processing.

15.4 Examine and manually classify a collection of short sound files

Suppose you have a number of short sound files with various types of sounds. For instance, Ishmael makes such files when you follow the procedure above, "*Automatically find calls in a large archive.*" One common thing to do with such files is to categorize them as different types of calls or types of noise.

For each category of sound you want to classify, create a new action using **Actions→Edit actions**. For each action, give it a unique hot key, check the box labeled **Go to the next file in a batch run**, go to the **Logging** tab, and check the **sound file name** and the **action's name** and/or **action's hot key** boxes. Uncheck the other boxes and click **OK**. Set your desired spectrogram viewing parameters. Then do **File→Open file**, select the files you wish to examine (use **Ctrl+A** to select all of the files in a directory), and click **Open**. (Note: Some earlier versions of Windows can't handle huge collections of files, like more than a thousand or two at a time.) Do **File→Save settings** if you so desire. Click **Run**. When a file appears, classify it by typing its hot key. When you do this, your classification for that sound file will get written to the log, and the next file in the batch run should appear ready for classifying.

15.5 Monitor a real-time sound signal

Connect your sound signal source to the sound input card you wish to use. In Ishmael, open the sound input card using **File→Sound card, File→National Instruments input**, or **File→Data Translation input**. Set the sampling rate and any other relevant parameters. Click **OK**. Repeat the steps in "*View a sound file to see what's in it*" above to enable equalization and disable other Ishmael options. Click the **Run** button. Adjust the time and frequency scaling until the sound scrolls across the screen at a reasonable pace. Do **File→Save settings** if you so desire.

To pick out calls as they appear on the scrolling display and save them for later analysis, simply select each call and do **File→Save selection as**. A faster way to do this is to make an action for each type of sound you find. Configure the action to save the current selection and unpause Ishmael. Then all you need do is select a call when it appears—Ishmael will automatically pause as soon as you click on the spectrogram—and press your action's hot key.

15.6 Record a real-time sound signal

Connect your sound signal source to the sound input card you wish to use. In Ishmael, open the sound input card using **File→Sound card, File→National Instruments board**, or **File→Data Translation board**. Set the sampling rate and any other relevant parameters. Click **OK**. Repeat the steps in "*View a sound file to see what's in it*" above to enable equalization if you want it and disable other Ishmael options. Choose **Record→Record sound** from the menu and specify the recording param-

eters you wish to use. Do **File→Save settings** if you so desire. Click the **Run** button to start the recording.

16.   Troubleshooting

When I try to run Ishmael, it says "A required library, BORLNDNM.DLL, is missing."

This occurs because you don't have the Borland libraries installed on your computer. You should get the normal Ishmael installation program from http://cetus.pmel.noaa.gov/cgi-bin/MobySoft.pl and re-run it.

When Ishmael starts up, it complains about a missing sound file.

This happens because the default settings file, which Ishmael loads at startup time, refers to files that are not present on your computer any more. To stop these warning messages, either (a) open a valid sound file (using **File→Open file...**), then do **File→Save settings as default** from the menu; or (b) edit the default settings file, C:\Program Files\Ishmael\ Ishmael 1.0\IshDefault.ipf, and remove the lines containing the offending file names. Be sure to save it as text only.

I changed the spectrogram or view parameters, but nothing happened.

Ishmael does not automatically recalculate spectrograms when you change the parameters. Click the **Run** button (the green right-arrow) to restart the spectrogram.

When Ishmael runs, all I see is a black rectangle.

This may happen because the brightness and/or contrast are not set correctly. Play with the scrollbars in the upper right corner until you can see a reasonable result. If you're analyzing a sound file, you may need to move the scroll bars and then click the **Run** button (the green right-arrow) again.

Another reason this can happen is because the time and/or frequency scaling are set badly, so all the data piles up in a very small area on the screen. Choose **View→Time scaling...** or **View→Frequency scaling...** from the menu and try other values.

See the next question too.

All I see are a bunch of vertical stripes, or solid-color areas.

This can happen when the amplitude and/or frequency scaling is set incorrectly. If amplitude scaling is set too small, you get vertical yellow stripes or just a bright yellow box in the signal waveform display. Choose **View→Amplitude range...** from the menu and adjust it. If the frequency scaling is set wrongly, you'll see colored vertical lines, or maybe just solid-color boxes, in the spectrogram display area. Choose **View→Frequency range...** from the menu and adjust the frequency range to be 0 to half your sampling rate. Note that the frequency values are in Hertz, *not* kilohertz.

This problem can also happen when the spectrogram floor and ceiling val-

ues are set poorly. To see if this is the problem, disable the floor and ceiling, and see if the problem persists. To correct it, adjust brightness and contrast until your calls appear in the spectrogram, then re-enable floor/ceiling with "automatic" checked.

I turned on equalization, and all that appears is a black (or white) box.
This is probably a brightness problem, caused by the equalizer's effect on the overall level of the spectrogram. Try changing the brightness scrollbar in the upper right corner of Ishmael's window.

I can see my spectrogram, but it has a wide black bar above and/or below it.
This happens when the frequency scaling is set badly. Choose **View→Frequency range** from the menu and adjust it to be from 0 to half the sampling rate of your sound.

Ishmael is not displaying as many channels of sound as I asked for.
Ishmael can be set up to input a certain number of sound channels (either from a file or from a real-time sound board), but display only some subset of them. To see all the channels, choose **View→All channels** from the menu.

Nothing shows up when I choose **Localize→Phone pair bearing** *from the menu.*
This is explained near the end of the Localization section, above.

My action's hot key isn't working.
Is the hot key in use elsewhere in Ishmael? Ishmael does not check for "hot key collisions"—the same key being used for two purposes. Check Ishmael's menus to see if your key appears on another menu item.

I have a Data Translation board, but it doesn't appear in Ishmael.
The most likely cause of this problem is that the file IshAcq.out is not in Ishmael's installation directory, C:\Program Files\Ishmael\Ishmael 1.0. Find IshAcq.out on your installation disk and place it in that directory.

*The file names I get when recording sound (***Record→Record sound***) or saving sound files (***Actions→Edit actions→(anything)→save the selection***) have the wrong time stamp.*
If you are using real-time input—a sound card or other data acquisition board—then Ishmael is getting the file's time stamp from Windows's idea of the time. To set the time in Windows, click the **Start** button at the lower left corner of the whole screen, then choose **Settings→Control Panels→Date/Time**.
If you are getting input from a file, it's a similar story. Most sound file types do not include a time stamp in the header of the file, and Ishmael isn't currently smart enough to figure out the file's time from, say, the file name. For files like this for which it can't determine a time, Ishmael pretends that the file starts at the same time that you click the **Run** button. You can always fool Ishmael by setting your computer's time to some desired time

and date.

Beamforming isn't working.

Make sure your input signal has at least three channels, and for file input, that all of the channels are enabled. Also, remember to configure the phone position file using **Localize→Loc options**.

A vertical black bar appears at the left edge of my spectrogram.

Each vertical strip of a spectrogram is time-aligned with the signal waveform in such a way that the center of the strip falls directly under the center of the group of samples in the waveform that are used to compute that strip. With a spectrogram hop size of $\frac{1}{2}$ or less, this requires that the very first spectrogram strip be offset a bit to the right of the start of the spectrogram display. This offset shows up as a black bar. Conversely, if the hop size is 2 or more, the first spectrogram strip has to be offset a bit to the left of the start of the spectrogram display.

The start of my detection function is not at time 0.

This is very similar to the reason why there is a vertical black bar at the left edge of the spectrogram. The detection function is arranged so that each point in it—each detection sample—is time-aligned with the center of the group of samples that went into computing it. This means that the first detection sample is after time 0.

The spectrogram ends before the signal waveform does.

At the end of the last spectrogram frame, there may be some samples left in the signal waveform—not enough sample to make another spectrogram frame, but a few. These show up in the waveform display, making it a bit longer than the spectrogram.

17.   Acknowledgments

18.   Conditions of Use

19. References

Bradbury, J.W., and S.L. Vehrencamp (1998): *Principles of Animal Communication.* Sinauer: Sunderland, MA.

Buck, J.R., H.B. Morgenbesser, and P.L. Tyack (2000): Synthesis and modification of the whistles of the bottlenose dolphin, Tursiops truncatus. *J. Acoust. Soc. Am., 108*(1), 407–416.

Charif, R.A., S.G. Mitchell, and C.W. Clark (1995): Canary 1.2 User's Manual. Technical report, Cornell Lab. Ornithol., Ithaca, NY.

Dooling, R.J., C.W. Clark, R. Miller, and T. Bunnell (1982): Program package for the analysis and synthesis of animal vocalizations. *Beh. Res. Methods and Instrumentation, 14*(5), 487.

Mellinger, D.K., and C.W. Clark (1997): Methods for automatic detection of mysticete sounds. *Mar. Freshwater Behav. Physiol., 29*(1), 163–181.

Mellinger, D.K., and C.W. Clark (2000): Recognizing transient low-frequency whale sounds by spectrogram correlation. *J. Acoust. Soc. Am., 107*(6), 3518–3529.

Van Trees, H.L. (1968): *Detection, Estimation, and Modulation Theory*, Vol. I. Wiley: New York.

Appendix: Keystrokes and Mouse Actions

Ishmael's predefined keystrokes:

Key	What it does
Space	Run; or if already running, Pause
Escape	Stop
DownArrow	Go to next batch file
UpArrow	Go to previous batch file
Alt+C	Log a comment
Alt+M	Execute Matlab command
Ctrl+O	Open file
Alt+P	Play current selection
Alt+S	Spectrogram parameters
Ctrl+S	Save selection as
Alt+W	Send current location to WhalTrak

Ishmael's mouse actions:

Mouse action	What it does
LeftClick	Remove current selection
LeftClick-and-drag	Make a selection
Shift+LeftClick	Extend the current selection
Ctrl+LeftClick	Trace a frequency contour
Shift+Ctrl+LeftClick	Do an Insta-Loc

If Ishmael is running when you click with the mouse in either the spectrogram or the signal waveform, it will pause.