

Getting Started

Chris Scott (C.D.Scott@leeds.ac.uk)

Download the project zip file and extract. The project directory contains a number of subdirectories,

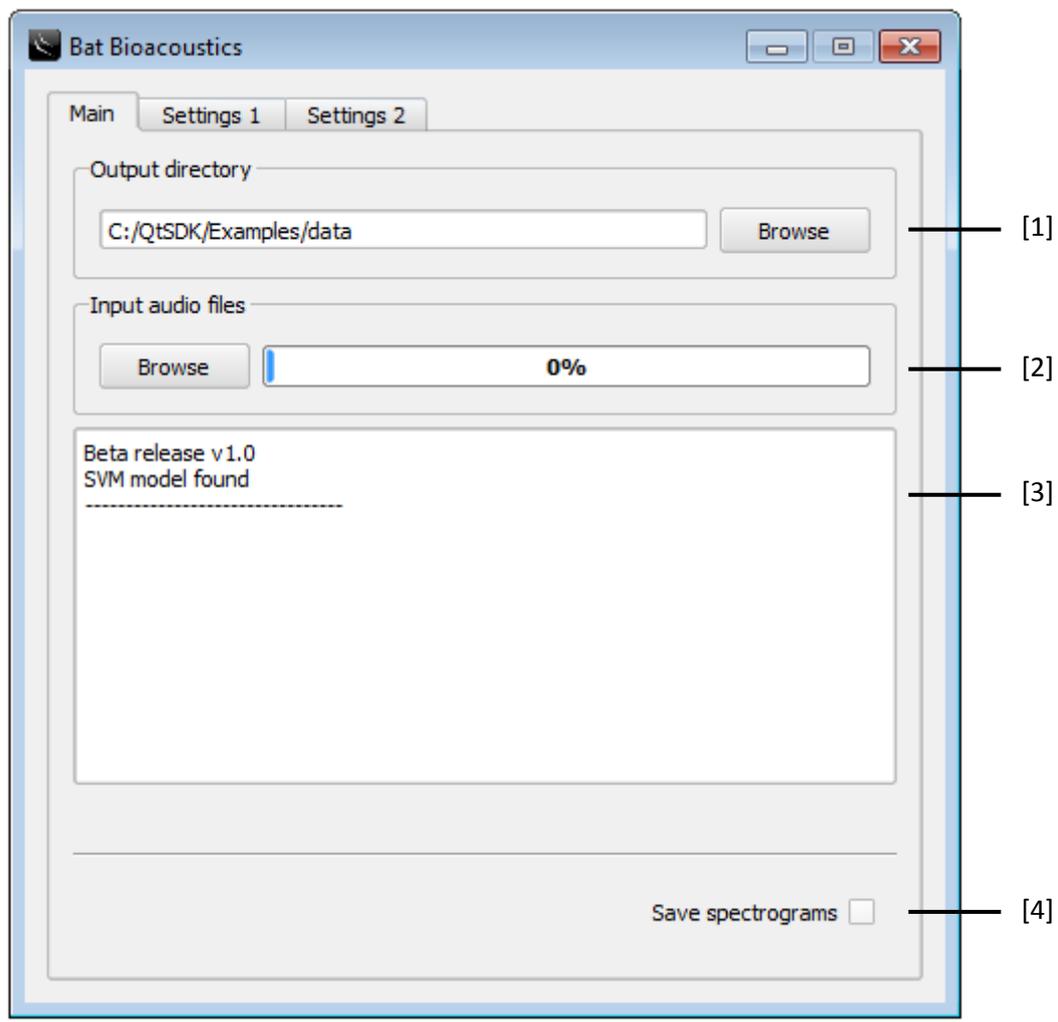
bin: contains the executable file and .dll files required by the program to run. These must be kept together in the same folder.

src: contains all the C++ source code files.

data: by default the analysis output files and spectrograms are saved here.

model: the program looks for the files required for support vector machine (SVM) -classification of calls here. If the appropriate files are not found, calls will not be classified but call features are still written to the call output file.

Main tab



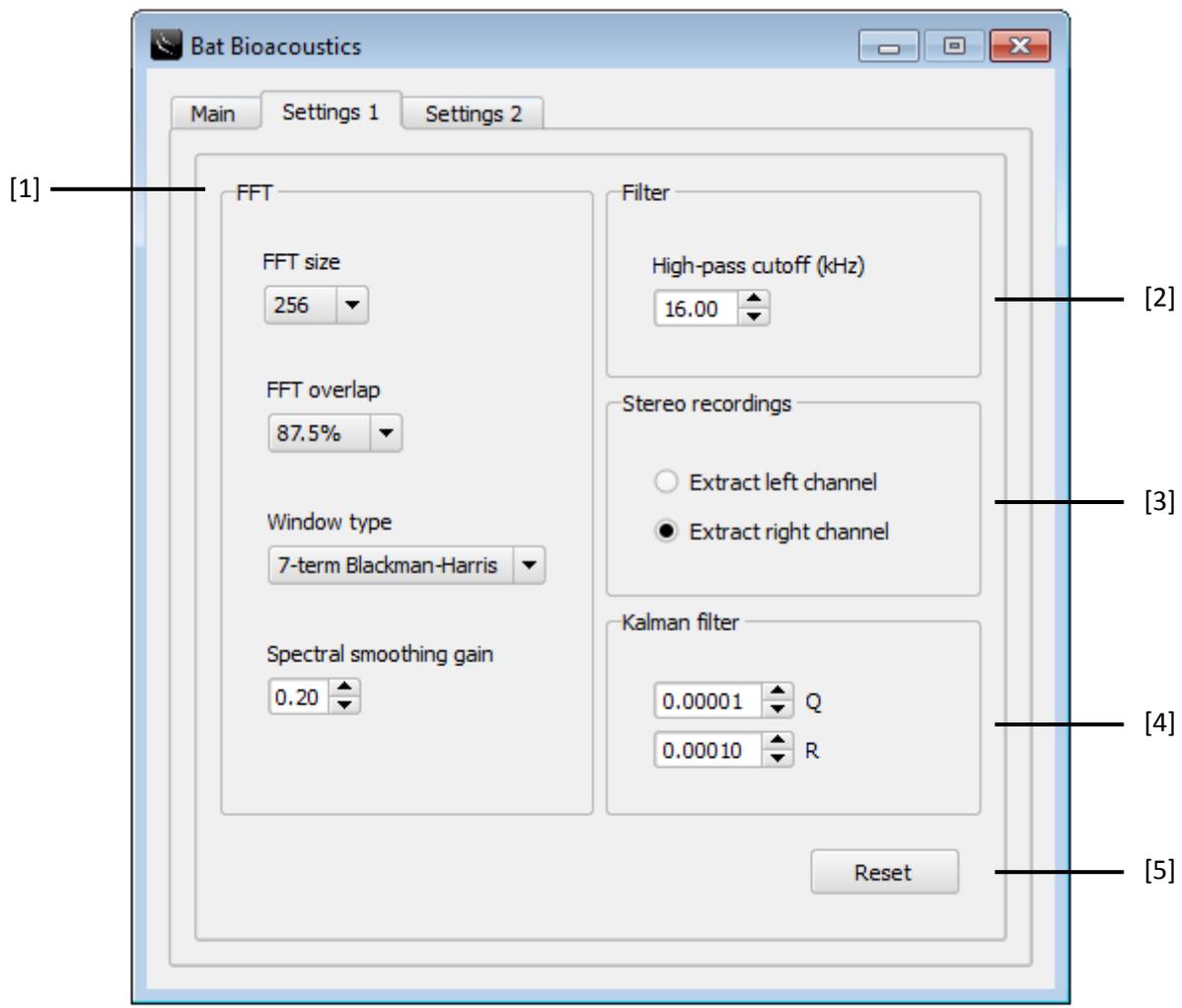
[1] The default directory for saving output files is in the data directory, the path to which is displayed in the text entry box. The browse button can be used to select an alternative directory for file saving.

[2] The input audio files browse button allows the selection of single or multiple audio files (WAV & FLAC) for analysis. Analysis begins immediately following file selection. Alternatively, the program supports drag and drop of files.

[3] The console window is used to report information to the user, for example files that failed to load.

[4] The 'Save spectrograms' checkbox provides the option for saving individual spectrograms of any extracted calls during analysis to the output directory. A 'spectrograms.html' file is created which can be opened by any modern web browser to view the spectrograms.

Settings 1 tab



[1] The program analyses audio files using the short-term Fourier transform (STFT). The process involves moving a fixed size analysis window through the file calculating the fast Fourier transform (FFT). Users can change the FFT window size (which influences the frequency resolution), the overlap between analysis windows (influencing the temporal resolution), the type of FFT window used, and the amount of spectral smoothing applied. Spectral smoothing is applied on the FFT spectrum using a backward-forward exponential smoothing algorithm. Higher gain values increase the amount of smoothing.

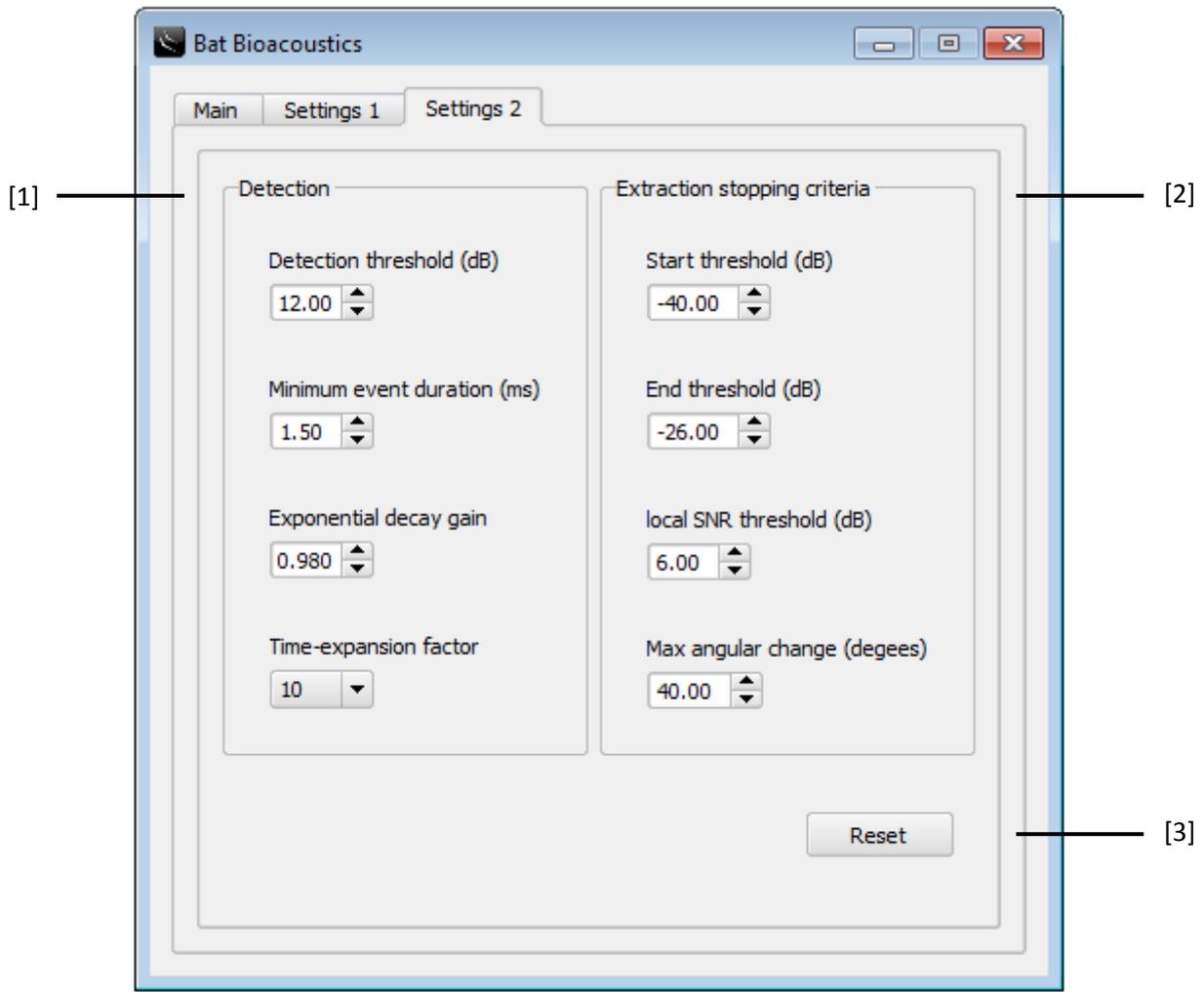
[2] A high-pass filter is used to reduce the influence of low-frequency noise on the analysis. Higher cutoff frequencies filter out progressively more of the audio spectrum.

[3] The user must select the channel to analyse from stereo recordings containing time-expanded and heterodyne signals.

[4] A discrete Kalman filter is applied to the tracked call frequencies. The Q and R noise covariance parameters influence the degree of smoothing.

[5] Changes made to user settings during the previous session are stored and applied on start-up, but can be returned to the default values using the reset button.

Settings 2 tab



[1] The level of the detection threshold influences the sensitivity of the call finding algorithm. Higher values increasingly leave quiet calls with low signal-to-noise ratios undetected. The minimum event duration is used by the program to ignore any extracted audio event shorter than the threshold. The exponential decay threshold sets the degree of temporal masking, designed to act as an echo filter, reducing the extraction of call echo. Higher gain increases the temporal masking effect. The time expansion factor is used to indicate the level of time expansion applied by the detector. The program assumes high sampling rate recordings are directly sampled by default, and ignores this setting.

[2] The call start and end thresholds are stopping criteria on extraction, specifying the level below the peak amplitude of the call that extraction should terminate. Low values for the call start threshold allow extraction to continue in calls that have suffered high frequency attenuation. Higher call end thresholds are useful in stopping call extraction prior to call echo. The local SNR threshold is a threshold designed to stop the extraction of calls as their level approaches that of the background noise. The final stopping criterion is a threshold on the maximum angular change between consecutive call frequency measurements. A high degree of angular change between analysis

frames indicates the extraction has jumped from the tracked harmonic into an upper harmonic, echo or noise.

[3] Changes made to user settings during the previous session are stored and applied on start-up, but can be returned to default values using the reset button.

Program output

If 'save spectrograms' was checked, for each extracted call, a spectrogram is generated and saved as a PNG image file (Figure 1).

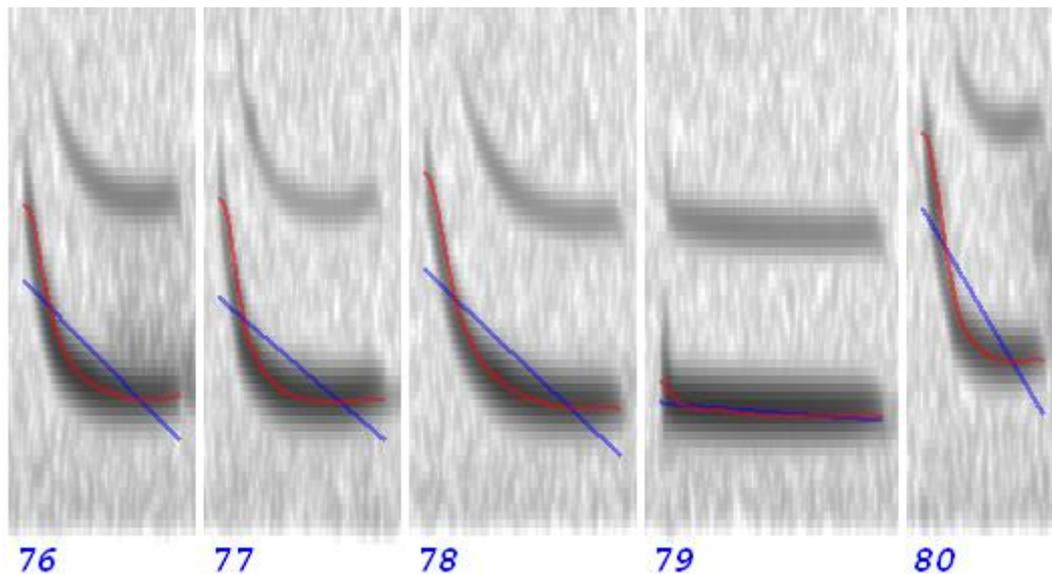


Figure 1 – Spectrograms output by the program. Numbers in the bottom left corner allow cross-referencing with the data in the 'call output' file. The red line traces the tracked and smoothed harmonic, and the blue line is the linear regression fit used by the feature extraction algorithm.

The 'call output.csv' file contains information from each individual extracted event, including call features and also classifications if an SVM model was provided. The 'recording output.csv' file summarises the classifications from each recording analysed.

Training a classifier

Support vector machine (SVM) classifiers can be created in R (<http://cran.r-project.org/bin/windows/base/>) using the package e1071 (Dimitriadou et al., 2011) and then used internally by the program during analysis. The first stage in building a classifier is to analyse a collection of reference calls. The 'call output.csv' file then needs to be edited before being imported into R, by filling in the 'Class' column to identify each extracted call (see example file 'training data.csv'). The 'train SVM.R' script can then be used to train and save a model and associated scale and label files, which should all be placed in the 'model' directory of the program. Next time the program is run the SVM model will be used to classify the extracted calls. We are

currently evaluating species-specific classification models that we have developed, and these will be made available on completion at <https://sourceforge.net/projects/batbioacoustics/>.

References

Evgenia Dimitriadou, Kurt Hornik, Friedrich Leisch, David Meyer and Andreas Weingessel (2011). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien. R package version 1.6. <http://CRAN.R-project.org/package=e1071>