PAM2Py: a brief guide

Orlando Camargo Rodríguez, Signal Processing Laboratory

March 28, 2021

1 Introduction

Passive acoustic monitoring is a non invasive ideal technique for listening to marine mammals, monitoring ocean soundscapes, earthquakes and evaluating anthropogenic noise harmful to marine life. Accurate ocean sound monitoring requires a continuous recording over vast ocean areas performed by different teams, often from different institutions and countries, each using its own set of equipment and recording devices. Specific recorders differ in hardware sensitivities, as well as gains and bandwidth filtering characteristics, and thus require proper calibration for a common reference in order to allow integration into a single coherent picture of ocean soundscapes. The PAMGuide software, written in Matlab [1] and available online, allows for such calibration [2,3]; the PAMGuide allows to process sound files from different sources and to introduce a proper compensation of hardware characteristics, providing a common reference for the calculation of sound pressure levels (SPLs). In order to achieve a full integration of SPL data into a common soundscape the PAMGuide has been updated within the JONAS project [4,5] to include a proper context in the form of EDF metadata to be exchanged between institutions. Even so code availability is not sufficient to support an efficient collaborative exchange between institutions since m-files require the purchase and renewal of Matlab licenses for all researchers, which for some participants can be problematic. The PAM2Py code [5] was developed in order to adress this issue: it incorporates all the capabilities of the original and extended PAMGuide, and it is written in Python [6], which can be dowloaded at no cost from the internet and can be installed in different OSes. This document briefly describes the PAM2Py GUI and showcases its usage.

2 General description

PAM2Py is a Python version of PAMGUide with the corresponding functionality [3] achieved by the call to standard bindings to Tcl/Tk, that provide buttons, input fields, etc., which gather information for Python functions from numpy and scipy to read files, process acoustic data, plot results, etc.; the GUI has been tested in Linux and Windows and the current version allows to

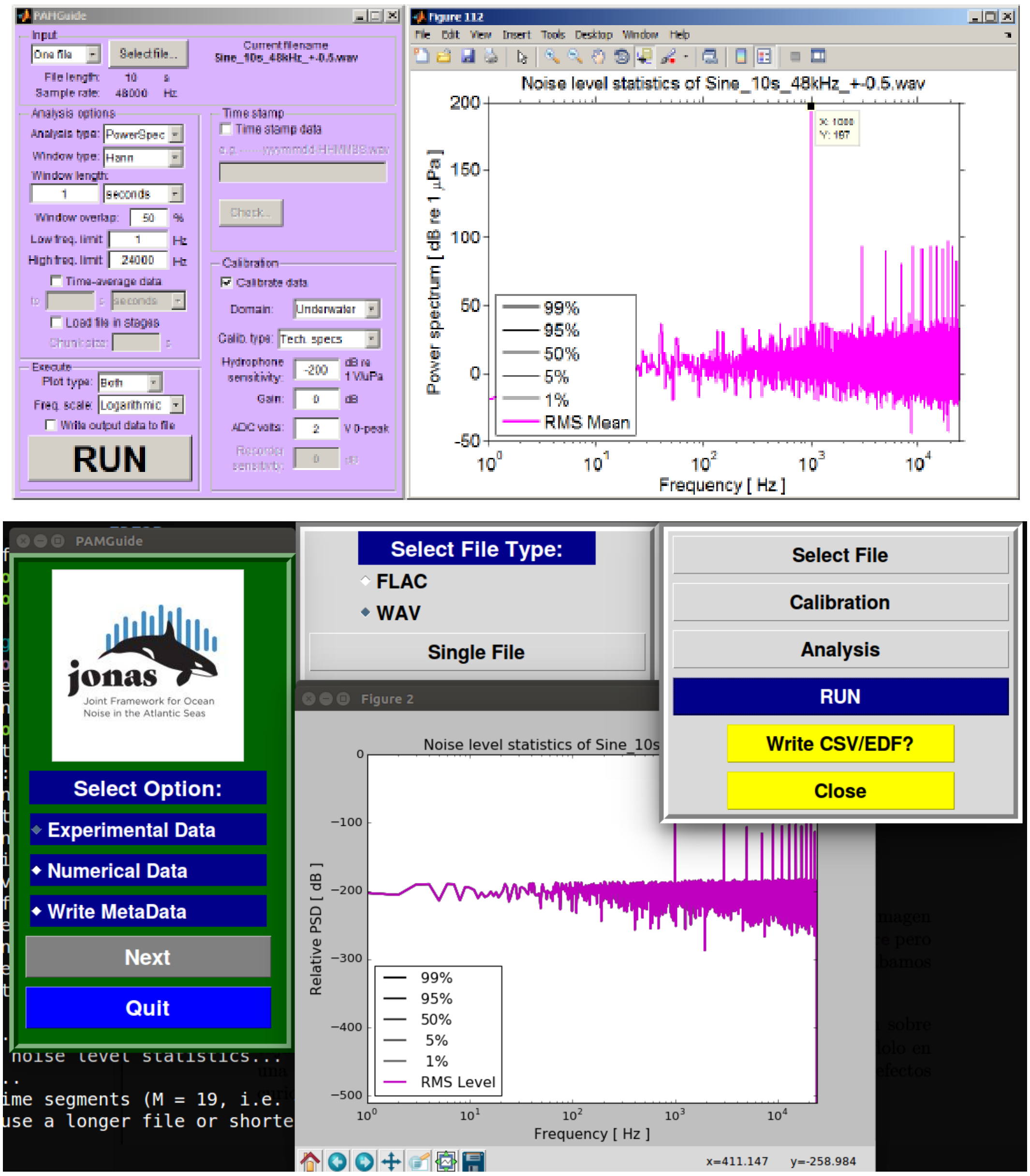
- Process *.WAV and *.FLAC audio formats.
- Process a single sound file.
- Process multiple sound files.
- Write CSV files.
- Write experimental our numerical metadata MET files.
- Write metadata EDF files.

The Python code is distributed with a pair of *.wav and *.flac files for test purposes. A pair of standalone functions to read and write EDF files has been developed too [5]. While the code has been tested extensively it is certainly not bug-free; improvements and suggestions can be addressed to the author via email (orodrig@ualg.pt)[1]. In contrast with the original Matlab GUI, which has different options for simultaneous calibration/analysis, single/batch processing, PAM2Py has been designed in order to guide the user through a set of simplified menus, which are expected to provide a logical sequence of selections (see Fig.1). Particular care has been taken to prevent multiple menus with identical options, concurrent menus, and to prevent function errors due to accidental processing without a proper definition of parameters. Input files for PAM2Py can be one of the following types:

- Experimental data: *.wav or *.flac files, for single or batch processing.
- Numerical data: Matlab *.mat files with SPL predictions calculated with an underwater acoustic model.

If not defined the PAM2Py GUI allows to write metadata *.met files. Output files can be of CSV or EDF type. Usage of the GUI will become clear in the description presented in the following sections.

[1] A GUI version written in R has been developed too, although with less capabilities and very little testing.



Input
One file | Select file...
Current filename
Sine_10s_48kHz_+-0.5.wav
File length: 10 s
Sample rate: 48000 Hz

Analysis options
Analysis type: PowerSpec
Window type: Hann
Window length: 1 seconds
Window overlap: 50 %
Low freq. limit: 1 Hz
High freq. limit: 24000 Hz
□ Time-average data
to s seconds
□ Load file in stages
Chunk size: s

Time stamp
□ Time stamp data
e.g. ......yymmdd-HHMMSS.wav
Check...

Calibration
☑ Calibrate data
Domain: Underwater
Calib. type: Tech. specs
Hydrophone sensitivity: -200 dB re 1 V/uPa
Gain: 0 dB
ADC volts: 2 V 0-peak
Recorder sensitivity: 0 dB

Execute
Plot type: Both
Freq. scale: Logarithmic
□ Write output data to file
RUN

```
noise level statistics...
..
ime segments (M = 19, i.e.
use a longer file or shorte
```

Figure 1 Top: PAMGuide GUI with a result of single file processing; bottom: PAM2Py GUI equivalent.

3 Experimental data

In Linux the PAM2Py GUI can be called from the command prompt with the command

```
python3 PAM2Py.py
```

which opens the GUI shown in Fig.2.

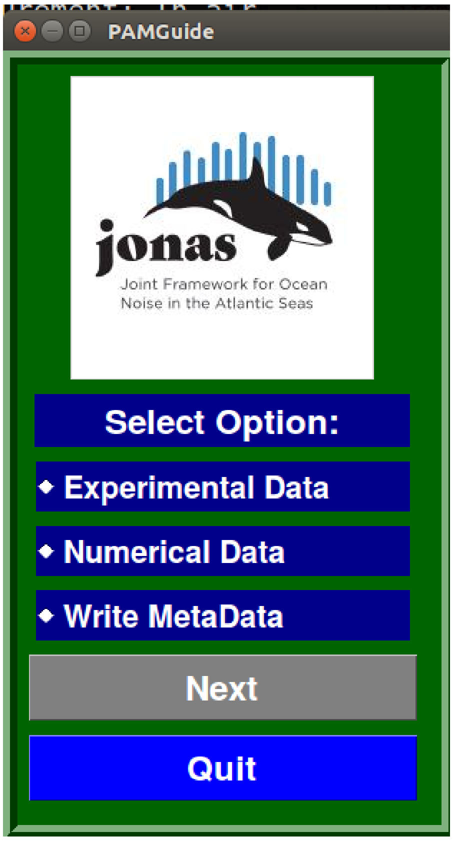


Figure 2 PAM2Py initial menu.

In the starting configuration the user will find a set of three options and two buttons; the Quit button ends (obviously) the GUI with a farewell message; clicking the Next button without a selected option triggers an alert; selecting the • Experimental Data option, followed by a click on the Next button opens a second menu, shown in Fig.3 (left), which can be closed (obviously) with the Close button, restoring the GUI to the initial configuration;

this button is present in every additional menu to prevent menu duplications; duplication is also avoided by disabling old menus when a new menu appears.

3.1 Single file processing

Selecting the ○ WAV option and clicking the Single File button opens a third menu, shown in Fig.3 (right).

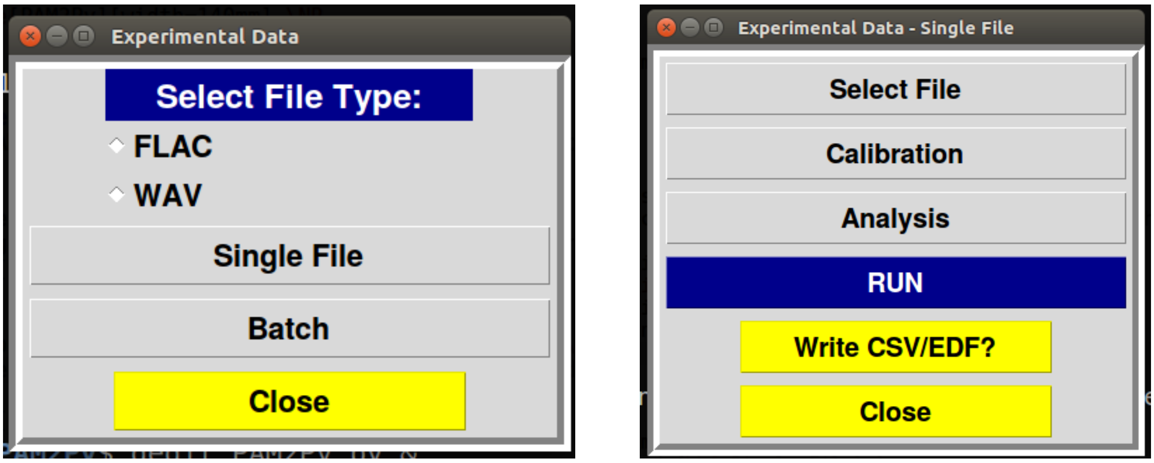


Figure 3 Left: PAM2Py second menu for experimental data; right: PAM2Py third menu for experimental data, single file processing.

The meaning of the buttons in the third menu is expected to be self-explanatory:

- Select File allows the selection of a *.wav file.

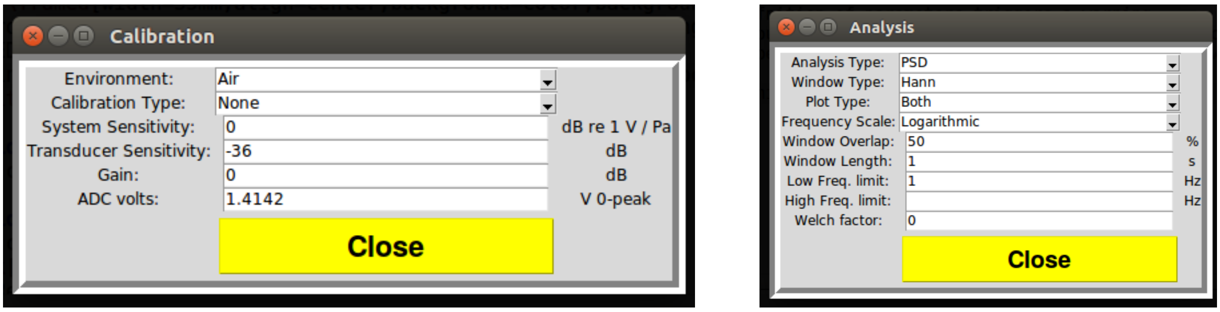
- Calibration opens a menu to input calibration parameters.

- Analysis opens a menu to input analysis parameters.

- RUN sends the input parameters to the Python functions to process the data and plot the results; clicking the button without a complete definition of parameters triggers a warning.

- Write CVS/EDF? opens a dialog asking if the results are to be written to a CSV or a EDF file; clicking the button before parameter definition and data processing triggers a warning.

Calibration and analysis parameters are described in detail in [3]; the corresponding menus are shown in Fig.4, they already contain reference values that can be changed by the user. Clicking the RUN button produces the requested results of data processing, and clicking the Write CVS/EDF? button allows to save the results in CSV or EDF format. In the case of EDF output the GUI expects the metadata to be defined in a *.met file. Writing of metadata files is discussed in section 5.



Environment: Air
Calibration Type: None
System Sensitivity: 0 dB re 1 V / Pa
Transducer Sensitivity: -36 dB
Gain: 0 dB
ADC volts: 1.4142 V 0-peak

Analysis Type: PSD
Window Type: Hann
Plot Type: Both
Frequency Scale: Logarithmic
Window Overlap: 50 %
Window Length: 1 s
Low Freq. limit: 1 Hz
High Freq. limit: Hz
Welch factor: 0

Figure 4 Left: experimental data, calibration parameters; right: experimental data, analysis parameters.

3.2 Batch file processing

Batch file processing is identical to single file processing, with the difference that it requires clicking on the Batch button in the menu shown in Fig.3 (left). The menu for batch processing is identical to the menu for single file processing, with the difference that it displays a Select Directory button to choose the directory containing all files of the chosen type (*.wav of *.flac). For batch processing to take place the sampling frequency of the different audio files should be the same.

4 Numerical data

Numerical data, stored in a Matlab *.mat file, can be processed by selecting in Fig.2 the ● Numerical Data option.

5 Writing Metadata

Experimental or numerical metadata *.met files can be created by selecting in Fig.2 the ● Write MetaData option. Details regarding the types of experimental and numerical metadata are described in [7].

References:

1. https://www.mathworks.com/products/matlab.html
2. N. Merchant, *Measuring acoustic habitats*, Methods in Ecology and Evolution (2015)
3. N. Merchant, *Measuring acoustic habitats, Appendix S1*, Methods in Ecology and Evolution (2015)
4. https://www.jonasproject.eu/
5. http://www.siplab.fct.ualg.pt/proj/jonas/pam2py.shtml
6. https://www.python.org/
7. Short User Manual for the Exchange Data Format package, JONAS project, November 2020.