JONAS
Joint Framework for Ocean Noise in the Atlantic Seas

PAM2Py users' manual

Project Name	JONAS
Project ID	INTERREG ATLANTIC - Contract EAPA 51/2018
Document name	PAM2Py users' manual
Document version	1.0
Authors	Orlando Rodríguez, Ricardo Duarte and Sérgio M. Jesus
Supervisor	JONAS
Date	April 2022
Status	PU (for Public)

Contents

1 Introduction

Ambient sound in the ocean is an important instrument to diagnose and monitor the ocean environment. The efforts to promote ocean preservation and the fulfillment of associated policies, among which the Marine Strategy Framework Directive (MSFD), led to requirements for wide maritime monitoring based on sound maps drawn over large geographical areas and across extended periods of time. Due to the ocean vastness, this goal can only be achieved through the cooperation between institutions and nations, agreeing to share data and knowledge relative to the oceanic areas under their jurisdiction.

However, different teams may use different equipments and may 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.

One way to make data exchange possible is first, eliminate the difficulties associated with data size, data description and data sensitivity and second establishing common procedure for determining averaged quantities and its statistics instead of raw data. In other words, a possible way to promote data exchange is to exchange averaged noise levels in frequency bands, along time intervals and sparse spatial grids instead of raw data.

In the frame of JONAS [1] project, financed by INTERREG Atlantic Area, PAM2Py was developed with the objective of facilitating the data exchange between institutions.

For this reason, PAM2Py is presented as an open source and open code tool that adopted the existing common standard processing steps described by Nathan *et al.* 2005 [2] to convert raw acoustic data into calibrated sound pressure levels considering different recording systems. PAM2Py allows the definition of several parameters as the definition of frequency bands, time sampling/averaging intervals, spatial grids, etc.

In other words, PAM2Py features averaged noise levels, standardized statistical indicators and holds both experimental, model-generated or mixed data. These various datasets are accompanied by metadata description fields, described in the Deliverable D4.3, that allow posterior data understanding, tracking and data visualization.

Written in Python, PAM2Py can be installed in different operating systems and is available for download at SIPLAB webpage and in GitHub platform. For extra documentation see the deliverable D4.3 of the JONAS project.

[1]Joint Framework for Ocean Noise in the Atlantic Seas

[2]Nathan et al. 2015, "Measuring acoustics habitats", Methods in Ecology and Evolution, 6:3, 257-265

2 General description

PAM2Py is a Python version of PAMGUide with the corresponding functionality achieved by the call to standard bindings to Tcl/Tk. PAM2Py gathers information for Python functions from *numpy* and *scipy* to read files, process acoustic data, plot results, etc. PAM2Py has been designed in order to guide the user through a set of simplified menus with self-explanatory menus, which are expected to provide a logical sequence of selections.

3 Description of the package contents

PAM2Py package includes multiple files organized by categories as described below:

1. **doc:** folder containing the necessary information to run PAM2Py;

   manual.pdf: brief PAM2Py user's guide;

   Metadata_Format.pdf: description of the metadata fields proposed by JONAS project;

2. **FLAC:** experimental test files in .FLAC format.

   Whistle

   DolphinGiggle

3. **Logos:** folder containing the JONAS project logo;

4. **MAT:** numerical test file in ".mat" format;

5. **MET:** folder containing two metadata files:

   metadata_numerical.met: numerical metadata test file;

   metadata_experimental.met: experimental metadata test file;

6. **WAV:** experimental test files in .WAV format.

   Sine_10s_48kHz_+-0.5

   WhiteNoise_10s_48kHz_+-0.5

7. **PAM2Py.ipynb**: is a PAM2Py's Jupyter notebook;

8. **py-files:**

   buffer.py: routine to generate a buffer array;

   LICENSE: GNU General public license;

   oct3dsgn.py: routine to design a one-third-octave filter;

   PAM2Py.py: main routine to call the PAM2Py GUI interface;

   PG_DFT.py: performs DFT-based analysis for PAMGuide.m;

   PG_Func.py: computes calibrated acoustic spectra from digital audio files;

   PG_TOL.py: performs 1/3-octave analysis using the standard filter bank method;

PG_Viewer.py: plots data analysed in PAMGuide.m;

PG_Waveform.py: performs pressure waveform analysis;

write_edf.py: writes the output in .h5 format;

read_edf.py: read the output in .h5 format;

9. **Others:**

   README: installing instruction file (IMPORTANT);

4 How to install

To install the PAM2Py package, the user should copy the complete downloaded folder to a preferable working directory. In case of a missing Python package/module, as for example *soundfile* module, use *pip* to install it according to the following examples:

- **Windows:** Using Anaconda prompt type: "pip install soundfile";
- **Linux:** In the Linux comandline type: "sudo pip3 install soundfile";
- **Conda:** In Conda platform type: "conda install -c bricew soundfile".

Note that is currently under development the implementation of PAM2Py in the JONAS VRE (Virtual Research Environment) through Jupiter notebooks in JupyterHub.

5 Getting started

This section will help you to get started with the PAM2Py. Each individual feature is described and explained using figures and photos where needed.

5.1 Graphical user interface

Launch your preferable Python application then open and run the "PAM2Py.py" file in the PAM2Py package. PAM2Py main window will be automatically displayed as shown in Figure 1. Note that: PAM2Py version is exhibit on the top left corner.

In the main window, users have three options depending on the nature of the data selected: write metadata, experimental data and numerical data. The first option, as the name suggests, is linked with the information regarding the data being used. The second option is related with processing real data obtained through experimental trials and the third option concerns the data resulting from models or from data assimilation

These options will be described in detail in the sections 5.2.1, 5.2.2 and 5.2.3 respectively. See the diagram presented in Figure 2 that corresponds to the PAM2Py working flow.

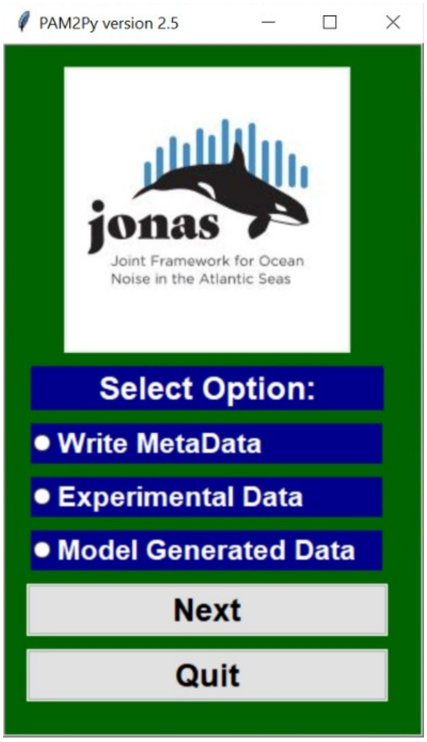


Figure 1: PAM2Py main graphical user interface.

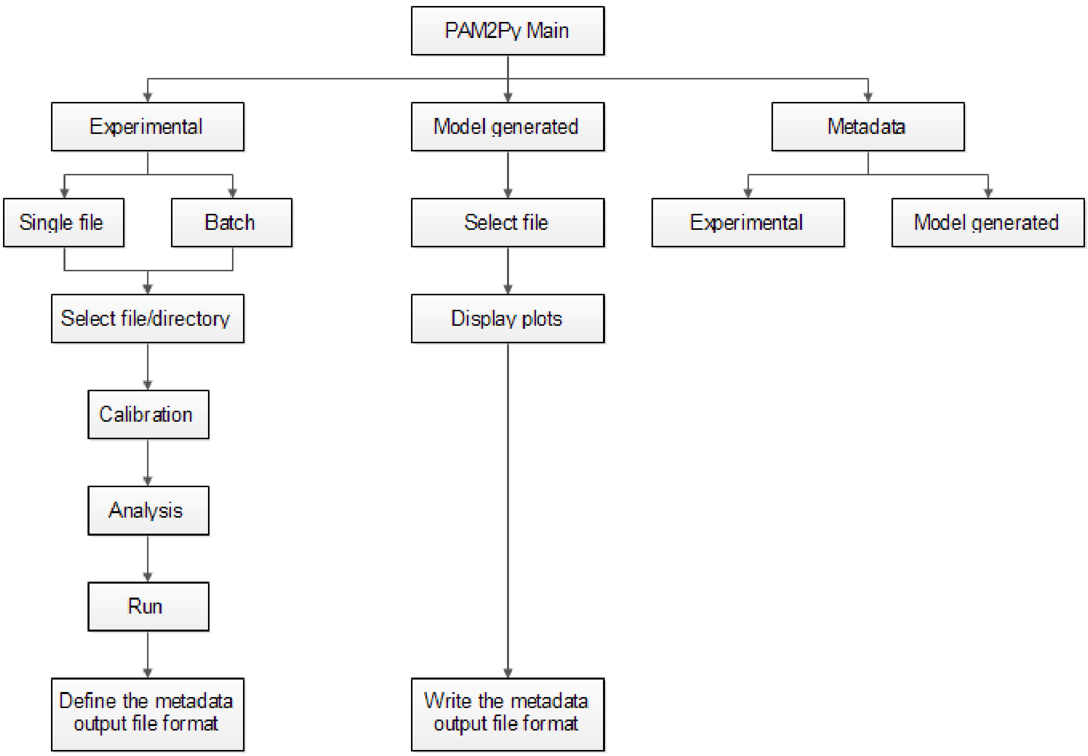
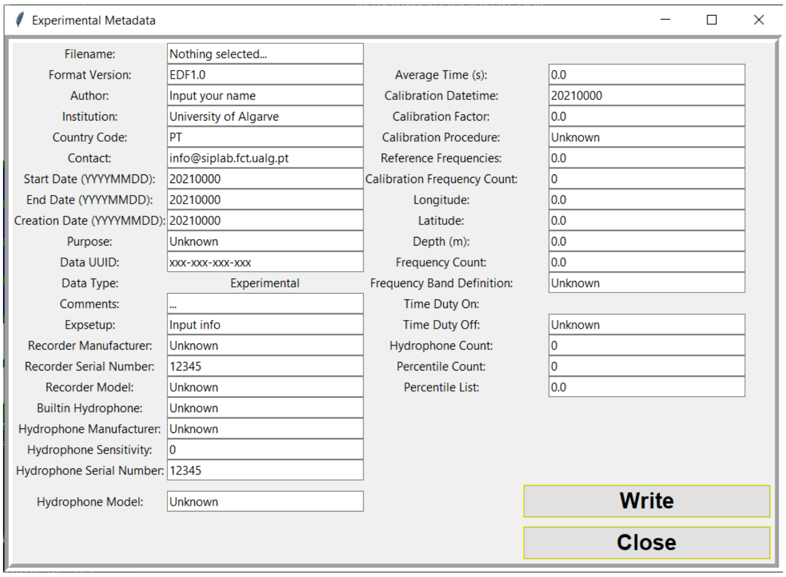
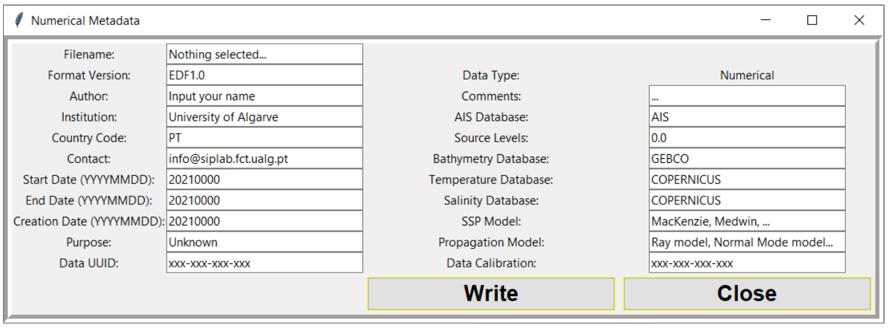


Figure 2: PAM2Py working diagram.



(a)



(b)

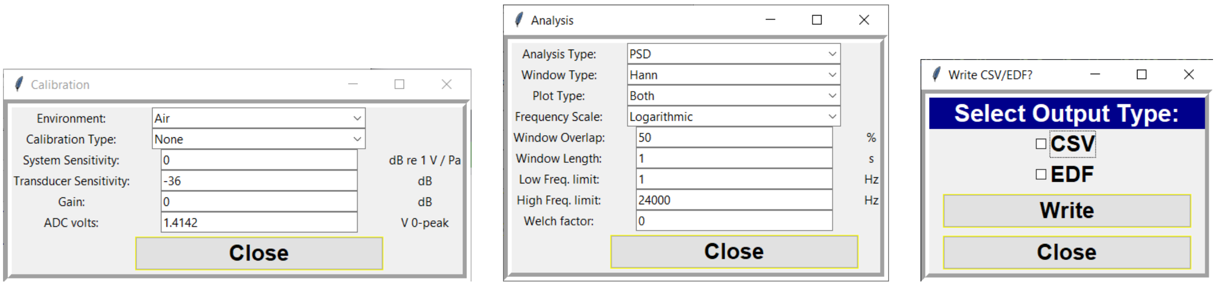
Figure 3: Experimental and model generated metadata windows.

5.2 Input data types

PAM2Py allows users to import two types of data depending on their needs and exchanging purposes. This data can be described throw several fields in the accompanying metadata. This section describes the data types considered in PAM2Py tool.

5.2.1 Metadata

The metadata window is displayed when the button "Write Metadata" is pressed which creates a `.met` file in your working directory. This window varies depending on the nature of the data being processed as showed in Figure 3 (a) and (b) for experimental and model generated metadata respectively. Fields in both windows are fully described in appendix section. `.met` files can be accessed using a typical ASCII writer/reader. Two example files are provided, "metadata_experimental.met" and "metadata_numerical.met" for experimental and numerical data respectively.



(a) (b) (c)

Figure 4: Experimental data processing settings: calibration (a), analysis (b) and metadata output (c).

5.2.2 Experimental data

The experimental input data contemplated is exclusively in non-compressed .WAV or .FLAC formats. Two options are available: users may process single files or complete directories using the "Batch" option. Since only raw data is considered, for this reason it is important to define the experimental setup as well as the specifications of the equipment used to record and the performed calibration. The processing steps were based on the previous processing steps fully described in Nathan *et al.* 2015 [3].

The processing steps are the same in both single or batch file processing. Figure 4 shows the experimental options that should be selected before processing the file(s). The ***calibration panel*** allows for the user to introduce the calibration parameters: domain (air or water), hydrophone sensitivity (dB re 1V/Pa), gain (dB), ADC volts (V zero-to-peak) and recorder sensitivity (dB) (Figure 4 (a)). The ***Analysis options panel*** integrates the selection of the analysis' type (PSD, TOL, Broadband, Waveform, PowerSpec and TOLf), the type of the Window (Hann, Rectangular, Hamming and Blackman), the window length, the window overlapping and frequency limits (Figure 4 (b)). The ***write CSV/EDF panel*** allows for the user to select the data export file formats which is only available after pressing the run button which starts the processing. If experimental data is being shared, two exporting formats are available: express format (creates a CSV file) or exchange data format (creates a HDF5 file) which respects the EDF file high level description format shown in Table 1. In the case where model generated data is being exchanged, only the EDF file will be available (Figure 4 (c)).

5.2.3 Model generated data

Model generated data is produced by numerical models as an attempt to forecast sound maps, often using AIS, wind, bathymetry and water column data relative to the area and time interval of interest. In this case, the input numerical file is tool dependent (.MAT only). Note that in this case the input data is already processed according to the data provider specifications. In this case, users need to select the input file and if needed create the plots.

[3]Nathan et al. 2015, "Measuring acoustics habitats", Methods in Ecology and Evolution, 6:3, 257-265

Note that in case of using model generated data it is necessary to previously create the .MAT-file (Py-file under development stage). For this reason, it is important to respect the following structure of variables:

1. total_number_of_grid_points;
2. longitude;
3. latitude;
4. depth;
5. frequency_band_definition;
6. frequency_count;
7. frequency;
8. time;
9. spl_values;
10. percentile_list.

These variables should follow the variable definition defined in the EDF Table 1 presented in the appendix section.

5.3 Output possibilities

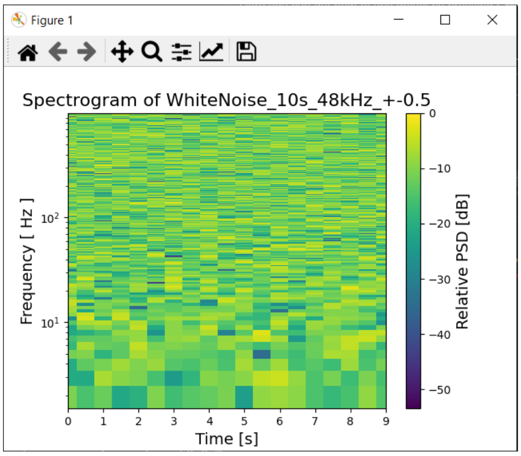
PAM2Py allows to obtain output files as `.csv`, `.h5` and `.met`. The `.csv` file contains data, the `.h5` file contains data and the respective metadata and finally the `.met` file only contains the metadata. Note that in case of using model generated data it is not possible to export the results in `.csv` format.

Additionally, the resulting `.csv` and/or the `.h5` and `.met` files will be created in the same directory of the input file. In the case of a single file being processed, the output file will be created with the same name as the raw (WAV/FLAC) data source file. Considering the processing of a complete folder (batch option), the output file will be created in a folder with the same name but with a "Batch" suffix. Note that if a `.csv` or `.h5` file exists in the same directory with the same name, it will be overwritten.

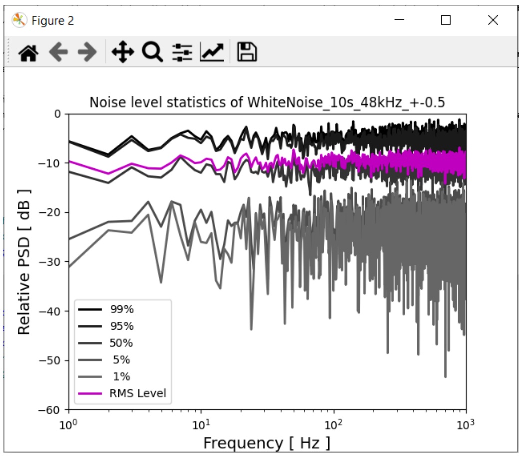
Beyond the possibility to obtain various output files, PAM2Py also generates two plots that correspond to the spectrogram and its statistics considering 1, 5, 50, 90 and 95 percentiles as well as the root-mean-square (RMS) shown in Figure 5 (a) and (b) respectively.

6 Contact

Any further explanation, debugging problems or suggestions should be transmitted to info@siplab.fct.ualg.pt.



(a)



(b)

Figure 5: PAM2Py output plots: spectrograms (a) and statistics (b).

A JONAS Exchange Data Format

Table 1: JONAS Project - Action 4.3: Exchange data format proposal using HDF5 file format.

Object name	Object field	Required	Object Type	Dimension	Field definition	Example
/	group				General description of the field	
format_version	attribute	Yes	string(10)		Format specification version	e.g."EDF1.0"
author	attribute	Yes	string(30)		Creator of the HDF5 file	e.g."Ricardo DUARTE"
institution	attribute	Yes	string(50)		Data provider Institution	e.g."University of Algarve, SIPLAB"
country_code	attribute	Yes	string(5)		Country and region codes according to ISO 3166	e.g."PT for Portugal"
contact	attribute	Yes	string(50)		Contact for all external queries in the future	e.g."e-mail:rjduarte@ualg.pt"
start_date	attribute	Yes	integer(20)		Data when the analysis started according to ISO 8601	e.g."20190131 for the 31st of January 2019"
end_date	attribute	Yes	integer(20)		Data when the analysis ended according to ISO 8601	e.g."20190231 for the 31st of February 2019"
date_of_creation	attribute	Yes	integer(20)		File creation date according to ISO 8601	e.g."20190310 for the 10th of March 2019"
purpose	attribute	Optional	string(500)		Objective of the experimental/numerical analysis	e.g."Evaluation of the underwater noise at Azores archipelago included in JONAS project"
data_uuid	attribute	Yes	string(30)		Data unique identification number, linking exchange data with raw data using: "country_code-year-month_day-datatype-file_number/total_files	e.g."PT-2020-0615-EXP-0001-0010"
data_type	attribute	Yes	string(20)		Indication whether is experimental, numerical or combined data	e.g."Combined"
comments	attribute	Optional	string(300)		Generic observations	e.g."Complete dataset"
/analysis_metadata	group				General description of the file metadata	
/experimental	group	Yes			General description of the experimental metadata	
setup	attribute	Optional	string(20)		Description of the deployment: Autonomous - AUT; Cable Mounted - CM; Combined - COMB; Bottom Frame - BF; From Vessel - FV; Glider - GLD Mooring with floating buoy - MFB; Other - OTH	e.g."CM" and "BF"
/recorder	group				General description of the recorder	
recorder_manufacturer	attribute	Optional	string(100)		Recorder manufacturer	e.g."MarSensing Lda."
recorder_serial_number	attribute	Optional	string(100)		Recorder serial number	e.g."SN45736"
recorder_model	attribute	Optional	string(100)		Recorder model	e.g."SR-1"
builtin_hydrophone	attribute	Optional	string(5)		Recorder and the hydrophone are one	e.g."Yes"
/hydrophone	group				General description of the hydrophones	
hydrophone_manufacturer	dataset	Optional	string(100)	hydrophone_count	Hydrophone manufacturer	e.g."MarSensing Lda."
hydrophone_sensitivity	dataset	Yes	float(10)	hydrophone_count	Hydrophone sensitivity provided by the manufacturer in $dBre1V/\mu Pa$	e.g."-185"
hydrophone_serial_number	dataset	Optional	string(100)	hydrophone_count	Hydrophone serial number	e.g."SN-SR1-2019-2"
hydrophone_model	dataset	Optional	string(100)	hydrophone_count	Hydrophone model	e.g."SR1"
/calibration	group				General description of the equipment calibration for each hydrophone	
calibration_frequency_count	dataset	Yes	integer(2)	hydrophone_count	Number of frequencies used to calibrate hydrophones	e.g."1"
calibration_datetime	dataset	Yes	integer(20)	hydrophone_count	Date and time when the system was calibrated according to ISO 8601	e.g."20180101 for the 1st of January of 2018"
calibration_factor	dataset	Yes	float(5)	hydrophone_count x calibration_frequency_count	Factor to convert raw data from volts to dB re $1\mu Pa$	e.g."1000"
calibration_procedure	dataset	Yes	string(300)	hydrophone_count	Procedure used to calibrate hydrophones according to ICES codes. See the entire ICES code list at: "vocab.ices.dk/?ref=1591"	e.g."CPC"
reference_frequencies	dataset	Yes	float(5)	hydrophone_count x calibration_frequency_count	Calibration frequencies in Hz	e.g."100"
/numerical_model	group	Yes			General description of the numerical model metadata	
ais_database	attribute	Yes	string(100)		Description of the AIS database used in the numerical models.	e.g."AISHub - www.aishub.net"
source_levels	dataset	Yes	integer(10)		Description of the emitted source levels in dB by vessel category according to their AIS ship type code First line indicating the ship type and second line the corresponding source level	e.g."[50 60 80; 170 120 180] "
bathymetry_database	attribute	Yes	string(100)		Description of the bathymetry database	e.g."GEBCO - www.gebco.net"
temperature_database	attribute	Yes	string(100)		Description of the water column temperature database	e.g."COPERNICUS - www.copernicus.eu"
salinity_database	attribute	Yes	string(100)		Description of the water column salinity database	e.g."COPERNICUS - www.copernicus.eu"
sound_speed_profile_model	attribute	Yes	string(100)		Description of model used to calculate the sound speed profile	e.g."K.Mackenzie-nine-term equation"
propagation_model	attribute	Yes	string(10)		Description of the propagation model	e.g."Kraken"
numeric_data_calibration	attribute	Yes	string(50)		Description of the experimental data file used to calibrate numerical models.	e.g."data_uidn:xxxx-xxxx-xxxx-xxxx"
/ambient_noise_dataset	group	Yes			General description of ambient noise	
/position	group	Yes			General description of the location	
hydrophone_count	dataset	Yes	integer(2)		Number of hydrophones	e.g."1"
total_number_of_grid_points	dataset	Yes	integer(10)		Total number of points in the numerical grid	e.g."1000"
longitude	dataset	Yes	float(10)	hydrophone/grid_count x time_count	Longitude coordinates vector in decimal degrees	e.g."[40.446,...,41.115]"
latitude	dataset	Yes	float(10)	hydrophone/grid_count x time_count	Latitude coordinates vector in decimal degrees	e.g."[79.982,...,81.281]"
depth	dataset	Yes	float(10)	hydrophone/grid_count x time_count	Depth vector according to the mean reference sea level in meters	e.g. "10"
/frequency	group	Yes			General description of ambient noise frequencies	
frequency_count	dataset	Yes			Number of frequencies	e.g."2"
frequency_band_definition	dataset	Yes	integer(10)	frequency_count	Describes the frequency band used and base	e.g."1/3-octave-band in base 2"
frequency	dataset	Yes	float(5)	frequency_count	Frequencies in Hz at which the SPL is calculated	e.g."63" and "126"
/time	group	Yes			General description of ambient noise time	
time_duty_on	attribute	Yes	integer(10)		Duration that the device is recording	e.g."30 minutes"
time_duty_off	attribute	Yes	integer(10)		Duration that the device is not recording. If the device is always recording type "0"	e.g."30 minutes"
time_count	dataset	Yes	float(10)		Number of time entrances since start_datetime	e.g."1250"
time	dataset	Yes	float(10)	time_count	Time vector in POSIX time seconds	e.g."[1562151105,...,1562153100]"
/sound_pressure_levels	group	Yes			General description of sound pressure levels (spl)	
averaging_time	attribute	Yes	integer(5)		Averaging time in seconds	e.g."1"
spl_values	dataset	Yes	float(10)	hydrophone/grid_count x frequency_count x time_count	SPL over time for all covered frequency bands in dB re $1\mu Pa^2/Hz$ Using: $SPL(f) = 10log_{10}\left(\frac{P_{ss}}{B\Delta f^2 p_{ref}^2}\right) - S(f)$	e.g."[25,...,30]"
/sound_pressure_levels_stats	group	Yes			General description of sound pressure levels statistics	
percentile_count	dataset	Yes	integer(2)		Number of percentiles	e.g."7"
percentile_list	dataset	Yes	integer(50)	percentile_count	List of percentiles	e.g. "5,10,25,50,75,90,95"
percentile_values	dataset	Yes	float(10)	hydrophone/grid_count x frequency_count x percentile_count	SPL considered percentiles in [dB re $1\mu Pa^2$]	e.g."[74,...,81]"