

Soundscapy: A python package for soundscape assessment and analysis

Andrew Mitchell¹
Bartlett School of Sustainable Construction
University College London, London, UK

ABSTRACT

Soundscape questionnaires are widely used to gather subjective information about people's perceptions and attitudes towards their acoustic environment. Despite the widespread adoption of ISO/TS 12913-3 guidelines for analyzing soundscape survey data, there are still several misapplications and interpretations. A recent study has proposed an innovative visualization approach for soundscape data analysis, using a probabilistic method that depicts the collective perception of a soundscape as a distribution of responses within the circumplex. To bring this method to life, a new open-source python package called Soundscapy has been developed. The main goal of this software is to enable the easy, accessible, and consistent application of analysis for soundscape data collected according to the ISO 12913 methods. This conference paper outlines the important features of Soundscapy, explains its basic functioning, lists its current capabilities, and gives recommendations for its best use. Finally, the future development of Soundscapy is proposed, including the integration of psychoacoustic analysis, predictive soundscape models, and soundscape indices for use in automated assessment and design.

1. INTRODUCTION

1.1. Overview of soundscape assessment and analysis

Soundscapy is a python toolbox for analysing quantitative soundscape data. Urban soundscapes are typically assessed through surveys which ask respondents how they perceive the given soundscape. Particularly when collected following the technical specification ISO 12913 Part 3 [1], these surveys can constitute a rich source of data for soundscape analysis. Soundscapy aims to provide a comprehensive set of tools for analysing such data, including functions for data validation, analysis, and visualization.

As proposed by Mitchell, Aletta, & Kang [2], in order to describe the soundscape perception of a group or of a location, we should consider the distribution of responses. Soundscapy's approach to soundscape analysis follows this idea, providing functions for visualizing the distribution of responses to soundscape questionnaires.

To support this work, international standards like [1] have been developed to provide a framework for the measurement, analysis, and reporting of soundscape perception. Soundscapy is a new open-source Python package that aligns with the requirements outlined in this standard,

¹andrew.mitchell.18\@ucl.ac.uk

aiming to make advanced soundscape analysis techniques accessible to a wide audience. By basing its core functionality on ISO 12913-3, Soundscapy aims to lower the barriers to entry and enable more people to conduct high-quality soundscape research.

1.2. Soundscape Assessment Framework in Soundscapy

At the core of Soundscapy are functions for validating, analysing, and visualising soundscape data.

Soundscapy's design is closely tied to the guidance provided in ISO 12901303

2. BACKGROUND

ISO 12913-3 is the international standard for soundscape surveys, which provides guidelines for the collection and analysis of soundscape data. The standard outlines a structured approach to soundscape data collection, including the use of questionnaires, sound recordings, and environmental measurements. The standard also provides a framework for the analysis of soundscape data, including the calculation of soundscape indices and the visualization of soundscape maps.

3. SOUNDSCAPY: HISTORY AND DESCRIPTION

Soundscapy is built on outcomes from the Soundscape Indices (SSID) project. The SSID project aimed to develop a new method for analyzing soundscape data that could provide more detailed insights into the perception of soundscapes. The SSID method uses a probabilistic approach to model the distribution of responses to soundscape questionnaires, allowing researchers to explore the relationships between different soundscape attributes and the overall soundscape quality.

[2]

3.1. Databases

Soundscapy was primarily developed to work with the International Soundscape Database (ISD) [3], which I will describe here and use for the examples in this paper. The ISD is a large database of soundscape recordings and survey data collected according to the SSID Protocol [4]. The database contains over 3,000 recordings, each with a corresponding survey that includes information about the soundscape, the location, and the respondents. The ISD is freely available to researchers and can be used to study a wide range of soundscape-related topics.

The ISD contains three primary types of data - surveys, pre-calculated psychoacoustic metrics, and binaural audio recordings. The surveys include several blocks of questions, the most important of which are the Perceptual Attribute Questions (PAQs). These form the 8 descriptors of the soundscape circumplex [5] - pleasant, vibrant, eventful, chaotic, annoying, monotonous, uneventful, and calm. In addition, each survey includes other information about the soundscape and demographic characteristics (age, gender, etc.). Finally, the survey section includes identifiers of when and where the survey was conducted - the LocationID, SessionID, latitude, longitude, start_time, etc.

The ISD is included with Soundscapy and can be loaded with the following code:

```
import soundscapy as sspy
df = sspy.isd.load()
df, excl_df = sspy.isd.validate(df)
df = sspy.isd.add_iso_coords(df)
```

3.2. Visualizing soundscape data

Visualizing soundscape data is crucial for exploring patterns and communicating findings. Soundscapy includes functions for creating

```
sspy.plotting.density(
    df.query("LocationID == 'CamdenTown'"),
    figsize=(4, 4),
)
```

3.3. Psychoacoustic Analysis

— Expand here —

This has been optimised for performing batch processing of many recordings, ease of use, and reproducibility. To do this, we rely on three packages to provide the analysis functions:

- Python Acoustics (acoustics): Python Acoustics is a library aimed at acousticians. It provides two benefits - first, the analysis functions are referenced directly to the relevant standard. Second, Soundscapy subclasses the Signal class to provide the binaural functionality, and any function available within the Signal class is also available to Soundscapy's Binaural class.
- scikit-maad [6] (maad): scikit-maad is a modular toolbox for quantitative soundscape analysis, focused on ecological soundscapes and bioacoustic indices. scikit-maad provides a huge suite of ecosoundscape focused indices, including Acoustic Richness Index, Acoustic Complexity Index, Normalized Difference Soundscape Index, and more.
- MoSQITo (mosquito): MoSQITo is a modular framework of key sound quality metrics, providing the psychoacoustic metrics for Soundscapy.

```
from soundscapy import Binaural
b = Binaural.from_wav(wav_dir.joinpath("CT101.wav"))
b.plot();
```

Figure 1:

The metrics currently available are:

- Python Acoustics : L_{Zeq} , L_{Aeq} , L_{Ceq} , SEL, and all associated statistics (L_5 through L_{95} , L_{max} and L_{min} , as well as kurtosis [7] and skewness).
- scikit-maad : So far, only the combined `all_temporal_alpha_indices` and `all_spectral_alpha_indices` functions from scikit-maad have been implemented; calculating them individually is not yet supported. `all_temporal_alpha_indices` comprises 16 temporal domain acoustic indices, such as temporal signal-to-noise ratio, temporal entropy, and temporal events. `all_spectral_alpha_indices` comprises 19 spectral domain acoustic indices, such as the Bioacoustic Index, Acoustic Diversity Index, NDSI, Acoustic Evenness Index, and Acoustic Complexity Index.

– MoSQITo : *cont.*

Soundscapy combines all of these metrics and makes it easy and (relatively) fast to compute any or all of them for a binaural audio recording. These results have been preliminarily validated through comparison of results obtained from Head Acoustics ArtemiS suite on a set of real-world recordings.

Table 1: Psychoacoustic metrics calculated by Soundscapy

```
metric = "LAeq"
stats = ("avg", 10, 50, 90, 95, "max")
label = "LAeq"
res_df = b.pyacoustics_metric(metric, stats, label)
res_df.round(2)
```

3.3.1. Consistent Analysis Settings

A primary goal when developing this library was to make it easy to save and document the settings used for all analyses. This is done through a `settings.yaml` file and the `AnalysisSettings` class. Although the settings for each metric can be set at runtime, the `settings.yaml` file allows you to set all of the settings at once and document exactly what settings were passed to each analysis function and to share these settings with collaborators or reviewers.

3.3.2. Defining Analysis Settings

Soundscapy provides the ability to predefine the analysis settings. These are defined in a separate `.yaml` file and are managed by Soundscapy using the `AnalysisSettings` class. These settings can then be passed to any of the analysis functions, rather than separately defining your settings as we did above. This is particularly useful when performing batch processing on an entire folder of wav recordings.

Soundscapy provides a set of default settings which can be easily loaded in:

```
analysis_settings = sspy.AnalysisSettings.default()
```

and the settings file for the Loudness calculation can be printed:

```
analysis_settings['MoSQITo']['loudness_zwtv']
```

These settings can then be passed to any of the analysis functions, like the following function which will calculate all of the metrics requested in the settings file:

```
res_df = b.process_all_metrics(analysis_settings, verbose=False)
```

3.3.3. Batch Processing

The other primary goal was to make it simple and fast to perform this analysis on many recordings. One aspect of this is unifying the outputs from the underlying libraries and presenting them in an easy to parse format. The analysis functions from Soundscapy can return a MultiIndex pandas DataFrame with the Recording filename and Left and Right channels in the index and a column for each metric calculated. This dataframe can then be easily saved to a .csv or Excel spreadsheet. Alternatively, a dictionary can be returned for further processing within Python. The key point is that after calculating 100+ metrics for 1,000+ recordings, you'll be left with a single tidy spreadsheet.

The Soundscape Indices (SSID) project for which this was developed has over 3,000 recordings for which we needed to calculate a full suite of metrics for both channels. In particular, the MoSQITo functions can be quite slow, so running each recording one at a time can be prohibitively slow and only utilize a small portion of the available computing power. To help with this, a set of simple functions is provided to enable parallel processing, such that multiple recordings can be processed simultaneously by a multi-core CPU.

```
ser_start = time.perf_counter()
for wav in wav_dir.glob("*.wav"):
    recording = wav.stem
    decibel = tuple(levels[recording].values())
    b = Binaural.from_wav(wav, calibrate_to=decibel)
    ser_df = add_results(df, b.process_all_metrics(analysis_settings,
    verbose=False))
ser_end = time.perf_counter()
```

```
par_start = time.perf_counter()
par_df = parallel_process(
    wav_dir.glob("*.wav"), df, levels, analysis_settings, verbose=False)
par_end = time.perf_counter()
```

In our initial tests on a 16-core AMD Ryzen 7 4800HS CPU (Fedora Linux 36), this increased the speed for processing 20 recordings by at least 8 times.

In testing, the MoSQITo functions are particularly slow, taking up to 3 minutes to calculate the Loudness for a 30s two-channel recording. When running only a single recording through, this has also been sped up by parallelizing the per-channel calculation, reducing the computation time to around 50s.

4. FUTURE DEVELOPMENT

In addition to continuing to improve the core functionality of Soundscapy, there are several areas where future development will be focused. Primarily, we aim to develop and integrate predictive soundscape models, which will allow us to predict the soundscape quality of a location based on its acoustic characteristics [8]. This will involve developing machine learning models that can predict soundscape quality based on acoustic features, such as sound levels, frequency content, and temporal patterns.

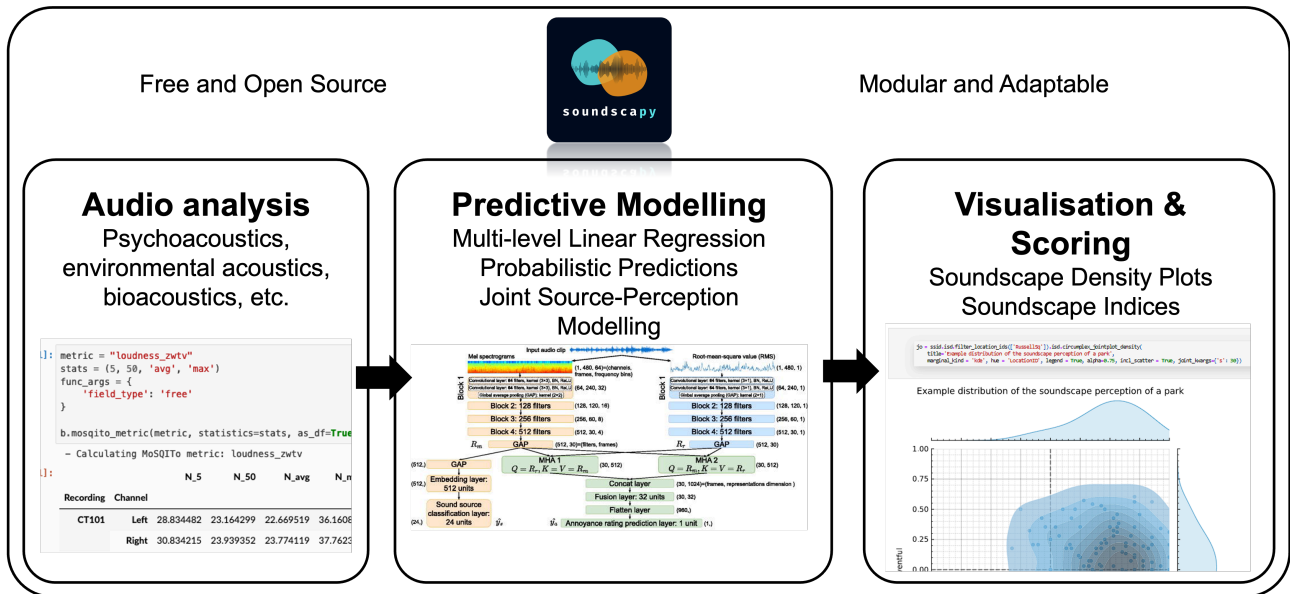


Figure 2: Future development plans for Soundscapy

5. CONCLUSIONS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Mi eget mauris pharetra et. Senectus et netus et malesuada fames ac. Nibh sed pulvinar proin gravida hendrerit lectus. Magna ac placerat vestibulum lectus mauris. In nulla posuere sollicitudin aliquam ultrices. Adipiscing bibendum est ultricies integer quis auctor. Mollis nunc sed id semper risus in. Neque laoreet suspendisse interdum consectetur libero id faucibus. Elit pellentesque habitant morbi tristique senectus et netus et. Pulvinar neque laoreet suspendisse interdum consectetur libero. Porta nibh venenatis cras sed. Tristique nulla aliquet enim tortor at auctor. Pellentesque diam volutpat commodo sed egestas egestas. Elit sed vulputate mi sit amet mauris.

Ipsum nunc aliquet bibendum enim facilisis. Mauris a diam maecenas sed enim ut sem. Et tortor consequat id porta nibh. Duis at consectetur lorem donec massa sapien faucibus et molestie. Nisl rhoncus mattis rhoncus urna neque viverra justo nec. Vulputate odio ut enim blandit. Nulla facilisi etiam dignissim diam quis enim. At augue eget arcu dictum varius duis at consectetur. Phasellus vestibulum lorem sed risus ultricies tristique nulla aliquet enim. Amet risus nullam eget felis eget nunc lobortis mattis. Sit amet nisl suscipit adipiscing. Pharetra et ultrices neque ornare aenean.

Molestie nunc non blandit massa enim. Volutpat diam ut venenatis tellus in metus vulputate eu scelerisque. Molestie at elementum eu facilisis sed odio. Hendrerit dolor magna eget est lorem ipsum dolor sit amet. Ipsum dolor sit amet consectetur adipiscing elit. Congue eu consequat ac felis donec et odio pellentesque. Proin nibh nisl condimentum id venenatis a. Molestie a iaculis at erat. Fermentum leo vel orci porta non pulvinar neque laoreet. Sed adipiscing diam donec adipiscing tristique risus nec feugiat in. Tellus cras adipiscing enim eu turpis egestas pretium. Cursus mattis molestie a iaculis at erat pellentesque adipiscing commodo. Elementum pulvinar etiam non quam lacus suspendisse. Gravida quis blandit turpis cursus in.

ACKNOWLEDGEMENTS

REFERENCES

- [1] ISO/TS 12913-3:2019, “Acoustics -- Soundscape -- Part 3: Data analysis”. 2019.
- [2] A. Mitchell, F. Aletta, and J. Kang, “How to analyse and represent quantitative soundscape data”, *JASA Express Letters*, vol. 2, no. 3, p. 37201, 2022, doi: [10.1121/10.0009794](https://doi.org/10.1121/10.0009794).
- [3] A. Mitchell *et al.*, “The International Soundscape Database: An integrated multimedia database of urban soundscape surveys -- questionnaires with acoustical and contextual information”. Zenodo, Oct. 2021. doi: [10.5281/zenodo.5578572](https://doi.org/10.5281/zenodo.5578572).
- [4] A. Mitchell *et al.*, “The Soundscape Indices (SSID) Protocol: A Method for Urban Soundscape Surveys--Questionnaires with Acoustical and Contextual Information”, *Applied Sciences*, vol. 10, no. 7, p. 2397, Apr. 2020, doi: [10.3390/app10072397](https://doi.org/10.3390/app10072397).
- [5] Ö. Axelsson, M. E. Nilsson, and B. Berglund, “A principal components model of soundscape perception”, *The Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. 2836–2846, 2010, doi: [10.1121/1.3493436](https://doi.org/10.1121/1.3493436).
- [6] J. S. Ulloa, S. Hauptert, J. F. Latorre, T. Aubin, and J. Sueur, “scikit-maad: An open-source and modular toolbox for quantitative soundscape analysis in Python”, *Methods in Ecology and Evolution*, vol. 12, no. 12, pp. 2334–2340, Sep. 2021, doi: [10.1111/2041-210x.13711](https://doi.org/10.1111/2041-210x.13711).
- [7] W. Qiu, W. J. Murphy, and A. Suter, “Kurtosis: A New Tool for Noise Analysis”, *Acoustics Today*, vol. 16, no. 4, p. 39, 2020, doi: [10.1121/at.2020.16.4.39](https://doi.org/10.1121/at.2020.16.4.39).
- [8] A. Mitchell, F. Aletta, T. Oberman, M. Erfanian, and J. Kang, “A conceptual framework for the practical use of predictive models and Soundscape Indices: Goals, constraints, and applications”, in *INTER-NOISE 2023 Conference*, Aug. 2023.