25-29 August 2024
inter.noise
NANTES FRANCE

Soundscapy: A python package for soundscape assessment and analysis

Andrew Mitchell[1]
Institute for Environmental Design & Engineering, The Bartlett
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 interpretations and challenges in application. To enable the easy, accessible, and consistent analysis of soundscape data, an open-source python package called Soundscape has been developed. This package implements a 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. In addition, functions for psychoacoustic and acoustic analysis of binaural data are included, with a focus on consistent and optimized processing of multiple recordings. 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 predictive soundscape models for use in automated assessment and design.

1. INTRODUCTION

Since 2018, the ISO 12913 series of standards has provided a framework for the measurement, analysis, and reporting of soundscape perception [1, 2]. The standard outlines a structured approach to soundscape data collection, including the use of questionnaires, binaural recordings, and environmental measurements. While other software tools exist for conducting soundscape analysis (see Section 4.4 for some examples), these are typically targeted towards audio analysis in the vein of soundscape ecology, rather than the analysis of soundscape perception data. Soundscapy aims to enable the analysis of human perceptual data from soundscapes, following the definition of 'soundscape' given in ISO 12913-1 [3].

In addition to implementing analysis methods for questionnaire data, Soundscapy also provides tools for the analysis of binaural recordings in line with ISO 12913-3. This includes the calculation of acoustic indices, such as the sound pressure level (SPL) and psychoacoustic indices, such as the loudness level (N) and sharpness (S). This paper will provide an overview of these features as implemented in the current version (v0.6), how these improve over existing tools, and a discussion of the future development of Soundscapy.

[1]andrew.mitchell.18@ucl.ac.uk

2. BACKGROUND

In 2010, Axelsson, Nilsson, & Berglund [4] proposed a *principal components* model of soundscape perception. Due to its similarity to the the widely-studied Russell's circumplex model of affect [5], Axelsson's principal component model is often referred to as the Soundscape Circumplex Model in soundscape literature. The circumplex model and the Swedish Soundscape Quality Protocol (SSQP) [6] utilizing it quickly became the predominant method of soundscape assessment in both scientific literature and professional practice [7], due to its ease of use, interpretability, and, crucially, its ability to summarise the complex interrelationships between soundscape descriptors within a straightforward and familiar two-dimensional space. Together with a similar principal component model in Cain et al. [8], the framework of the circumplex model of soundscape perception was subsequently adapted into an integral part of the standardised data collection, specifically in Method A of ISO/TS 12913-2 [1].

The soundscape circumplex model is composed of eight scales, which closely resemble the eight scales of the Circumplex Model of Affect [5]. The 8 scales are arranged in two bipolar dimensions, with pleasant-annoying along the x axis (valence) and eventful-uneventful along the y axis (arousal). These scales can be projected into the two dimensional circumplex space using the following equations [9]:

$$P_{ISO} = \frac{1}{\lambda_{Pl}} \sum_{i=1}^{8} \cos\theta_i \cdot \sigma_i \tag{1}$$

$$E_{ISO} = \frac{1}{\lambda_{Pl}} \sum_{i=1}^{8} \sin\theta_i \cdot \sigma_i \tag{2}$$

where i indexes each circumplex scale, θ_i gives the angle for the circumplex scale, and σ_i is the value for that scale. The $1/\lambda$ provides a scaling factor to bring the range of P_{ISO}, E_{ISO} values to $[-1, +1]$:

$$\lambda_{Pl} = \frac{\rho}{2} \sum_{i=1}^{8} |\cos\theta_i| \tag{3}$$

$$\lambda_{Ev} = \frac{\rho}{2} \sum_{i=1}^{8} |\sin\theta_i| \tag{4}$$

where ρ is the range of the possible response values (i.e. 4 for the Likert responses used in ISO/TS 12913-3).

Currently, the soundscape community relies heavily on the framework proposed in ISO/TS 12913-2, both for theory development and for procuring empirical evidence of the benefits of the soundscape approach in real life scenarios. In a recent literature review, Aletta & Torresin [7] identified 254 scientific publications which have referred to ISO 12913 since its publication in 2018, with 50 of them appropriately making use of the data collection methods. Of those, several papers included multiple studies, with 51 studies making use of the circumplex model as recommended in the ISO standard. In addition, the circumplex model has been used in many more studies without reference to the ISO standard [10].

However, there is currently no standardised software enabling the circumplex analysis recommended by the ISO 12913 standards. This can lead to inconsistencies, errors, and delays in the application of the recommendations. With the revision of ISO 12913 Part 3 currently underway and proposals for more advanced analysis methods being discussed, an easy-to-use and free tool is even more necessary to ensure the consistent and validated application of the standard.

In addition to the recommendations for the analysis of this questionnaire data, the ISO/TS 12913 series of standards also provides a framework for the analysis of binaural recordings in ISO/TS 12913-3. This includes the calculation of acoustic indicators, such as the sound pressure level (SPL) and psychoacoustic metrics, such as the loudness level (N) and sharpness (S). These indicators 'enable the characterization of the acoustic environment [...], the quantification of the acoustical impact on the listener, and the exploration of relationships between physical properties of the environments and human response behaviour' [2].

Several software tools exist for psychoacoustic analysis, the most notable of which is the ArtemiS suite from HEAD Acoustics [11]. These commercial software packages are widely used in the industry for the analysis of sound and vibration data. However, these tools are often prohibitively expensive for many purposes and are not specifically designed for the analysis of large scale soundscape data. The recent development of the MoSQITo sound quality library [12] has provided a free open-source python alternative for single-channel psychoacoustic analysis. Soundscapy builds upon MoSQITo, to implement the psychoacoustic analysis methods recommended in ISO/TS 12913-3 for binaural recordings.

3. HISTORY AND DESCRIPTION

Soundscapy was initially developed for working with data collected as part of the Soundscape Indices (SSID) project [13]. A method for *in situ* data collection was developed, named the SSID Protocol, which enabled simultaneous collection of binaural recordings, survey questionnaires, and environmental data [14]. The SSID Protocol was designed to be compatible with the ISO/TS 12913 standards, optimized for large scale data collection. The data collected under this protocol forms the open-access International Soundscape Database (ISD) [15], which is described further in Section 4.1.

Building upon this database, a new method for analysing soundscape circumplex data was developed by Mitchell, Aletta, & Kang [16]. This method uses a distribution-based approach to model the distribution of responses to soundscape questionnaires within the circumplex. In this method, each response is transformed according to Equation 1 and Equation 2, and the distribution of these responses in the circumplex is visualized using a kernel density estimation. In this way, the collective perception of a group or of a location can be thought of as the bivariate distribution of responses within the circumplex. Soundscapy was initially developed as the research code to explore and develop this method in [16] before being published as an open-source package in 2022.

4. FEATURES

Soundscapy is an open package of Python functions. This means there is not a user interface and it is necessary to install and write Python code. The functions are designed to be simple to use even with limited experience writing code.

4.1. Databases

Soundscapy was primarily developed to work with the International Soundscape Database (ISD)[2] [15], which will be described here and used for the examples in this paper. The ISD is a large database of soundscape recordings and survey data collected according to the SSID Protocol [14]. The database contains 2,706 recordings, paired with 3,590 survey responses.

[2]The ISD is available from Zenodo https://zenodo.org/records/10672568.

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 [4] - 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()
```

The final bit of information for the survey is the `GroupID`. When stopping respondents in the survey space, they were often stopped as a group, for instance a couple walking through the space would be approached together and given the same `GroupID`. While each group completes the survey, a binaural audio recording is taken, typically lasting about 30 seconds. Therefore, each `GroupID` can be connected to around 1 to 10 surveys, and to one recording.

In addition to the ISD, Soundscapy includes a module for working with the Soundscape Attributes Translation Project (SATP) dataset[3] a project to provide validated translations of soundscape attributes in languages other than English. The SATP includes 19,089 survey responses, including 708 participants, in 19 languages. It can be loaded within soundscapy using `sspy.datasets.satp.load_zenodo()`. In addition, work is currently underway to implement a module for working with the Affective Responses to Augmented Urban Soundscapes (ARAUS) dataset [17].

4.2. Working with soundscape questionnaire data

All data in Soundscapy is handled using the `pandas` library [18]. Soundscapy implements some built in functions for working with both the ISD specifically and with soundscape data more generally. One example of an ISD specific function is `isd.validate()` which implements a series of data quality checks on the questionnaire data [19] and returns a validated dataset `df` and a set of the excluded data `excl_df`. `add_iso_coords()` implements the extended versions of the ISO 12913-3 method for calculating the coordinates within the circumplex from the perceptual attributes, given in Equation 1 and Equation 2, and adding them to the dataframe under the keys `ISOPleasant` and `ISOEventful`.

```
df, excl_df = sspy.isd.validate(df)
df = sspy.surveys.add_iso_coords(df)
```

Soundscapy expects the PAQ values to be Likert scale values ranging from 1 to 5 by default, as specified in ISO/TS 12913-2 and the SSID Protocol. However, it is possible to use data which, although structured the same way, has a different range of values. For instance, this could be a 7-point Likert scale, or a 0 to 100 scale. By passing these numbers to the `add_iso_coords()` function as `val_range=(0, 100)`, Soundscapy will automatically scale the ISOCoordinates from -1 to $+1$ according to Equation 3 and Equation 4.

[3]The SATP dataset is available from Zenodo https://zenodo.org/records/10159673.

4.3. Visualizing soundscape data

Once the data is loaded, it can be filtered as needed and visualise the distribution of responses within the soundscape circumplex. The plotting functions make use of the seaborn `kdeplot()` [20], adding additional customizations and features.

```
sspy.plotting.jointplot(
  sspy.isd.select_location_ids(df, "PancrasLock")
)
sspy.plotting.density(
  sspy.isd.select_location_ids(df, ("CamdenTown", "PancrasLock")),
  title="Comparison between two soundscapes",
  figsize=(5, 5),
  hue="LocationID",
  density_type='simple'
)
```

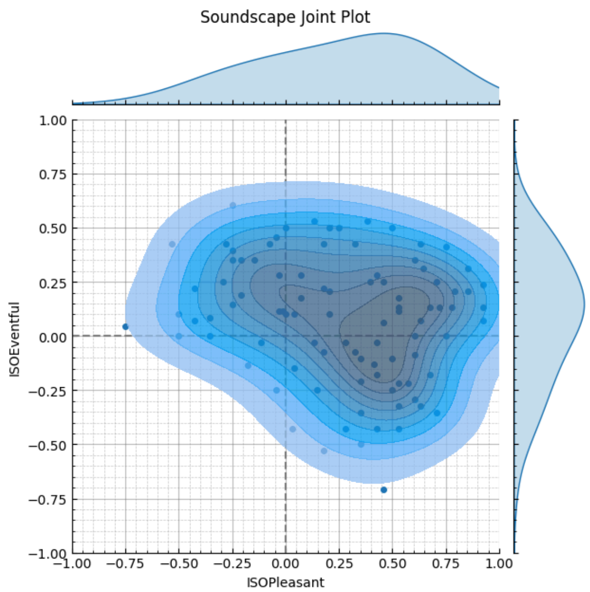

Figure 1: A jointplot including the joint and marginal distributions.

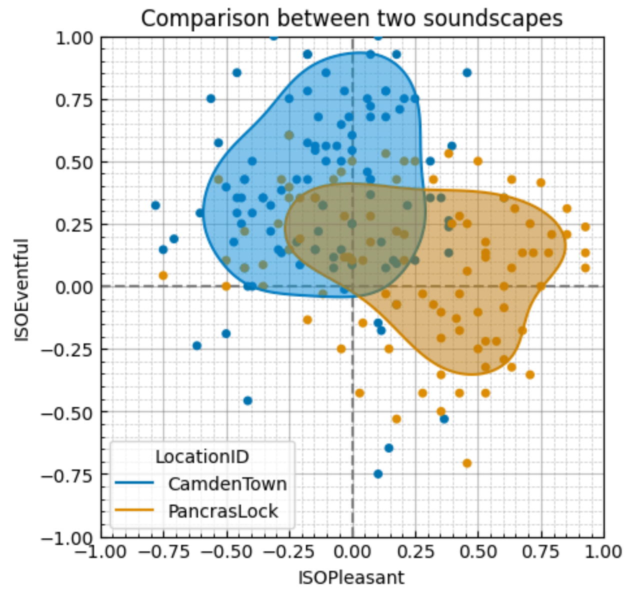

Figure 2: A simple density plot of two soundscapes.

4.4. Psychoacoustic, Acoustic, and Bioacoustic Analysis

To implement the binaural recording processing, we rely on three packages:

- 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 [21] (`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 (`mosqito`): 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()
```

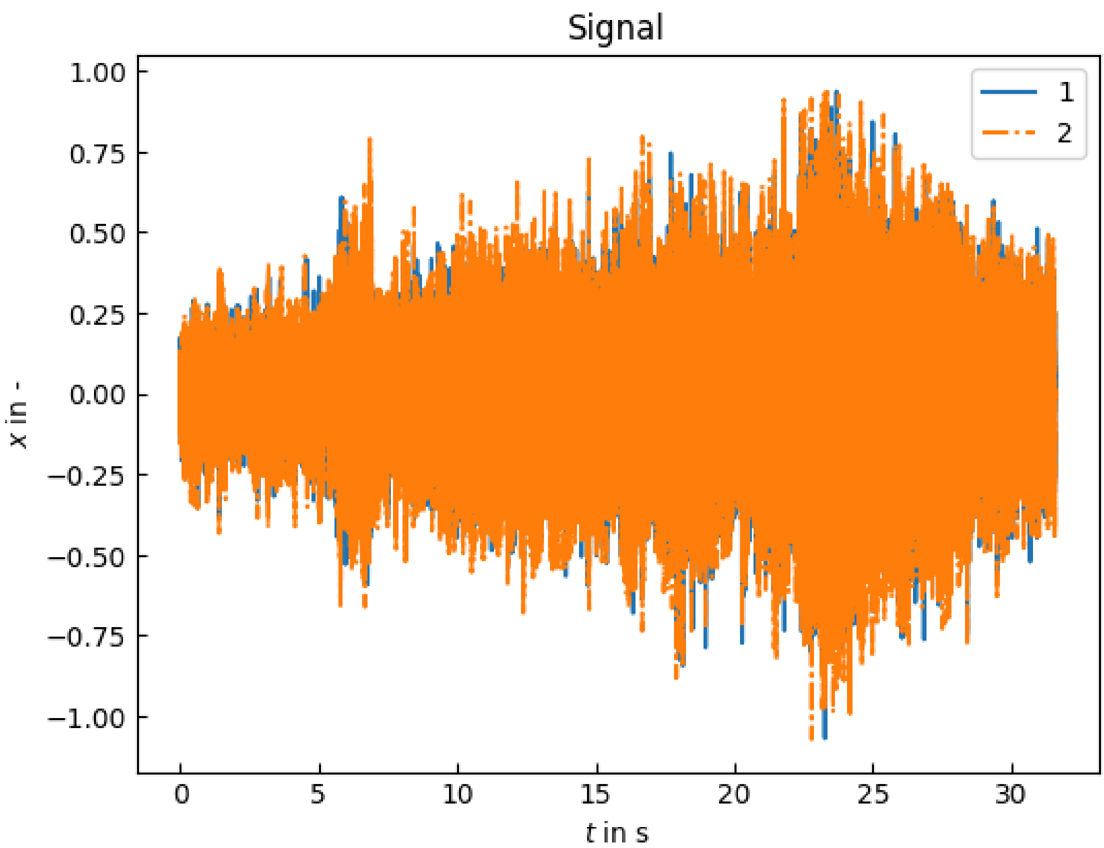

Figure 1: Loading and viewing a binaural recording in Soundscapy.

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 [22] 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_spectra_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: Zwicker time-varying Loudness (N) [23] and Roughness (R) [24] are implemented directly. In addition, Sharpness (S) can be calculated directly or from the the loudness parameters.

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.

```
metric = "loudness_zwtv"
stats = (5, 50, 'avg', 'max')
func_args = {'field_type': 'free'}
b.mosqito_metric(metric, statistics=stats, verbose=False, func_args=func_args)
```

Table 1: Direct output of psychoacoustic metrics calculated by Soundscapy for a single binaural recording.

		N_5	N_50	N_avg	N_max
Recording	Channel				
Rec	Left	45.126243	36.826403	35.434689	47.617004
	Right	47.388868	38.385263	37.061130	49.509641

4.4.1. 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.

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 e.g. the Loudness calculation can be printed:

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

```
{'run': False,
 'main': 5,
 'statistics': [10, 50, 90, 95, 'min', 'max', 'kurt', 'skew', 'avg'],
 'channel': ['Left', 'Right'],
 'label': 'N',
 'parallel': True,
 'func_args': {'field_type': 'free'}}
```

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)
```

4.4.2. 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, as shown in Table 1. 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.

When processing many files, the MoSQITo functions in particular 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. To demonstrate the performance improvement, we can process a set of 20 test recordings in series and in parallel.

```
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)
    )
```

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

Tested on a Macbook Pro M2 Max, processing 20 recordings (total of 10 minutes, 41 seconds of audio) in series took 29.4 minutes, while processing the same 20 recordings in parallel took 7.7 minutes, a speed up of 3.8 times.

5. FUTURE DEVELOPMENT AND CONCLUSIONS

In addition to continuing to improve the core functionality of Soundscapy, there are several areas where future development will be focused. As of v0.6, Soundscapy is primarily focused on implementing the analysis called for in ISO/TS 12913-3. Primarily, we aim to integrate new and existing predictive soundscape models, which will allow us to predict the soundscape quality of a location based on its acoustic characteristics [25]. The goal is to develop Soundscapy into a robust modelling and prediction pipeline, which allows for the estimation of soundscape perception based on quantifiable or estimable factors (see Figure 2).

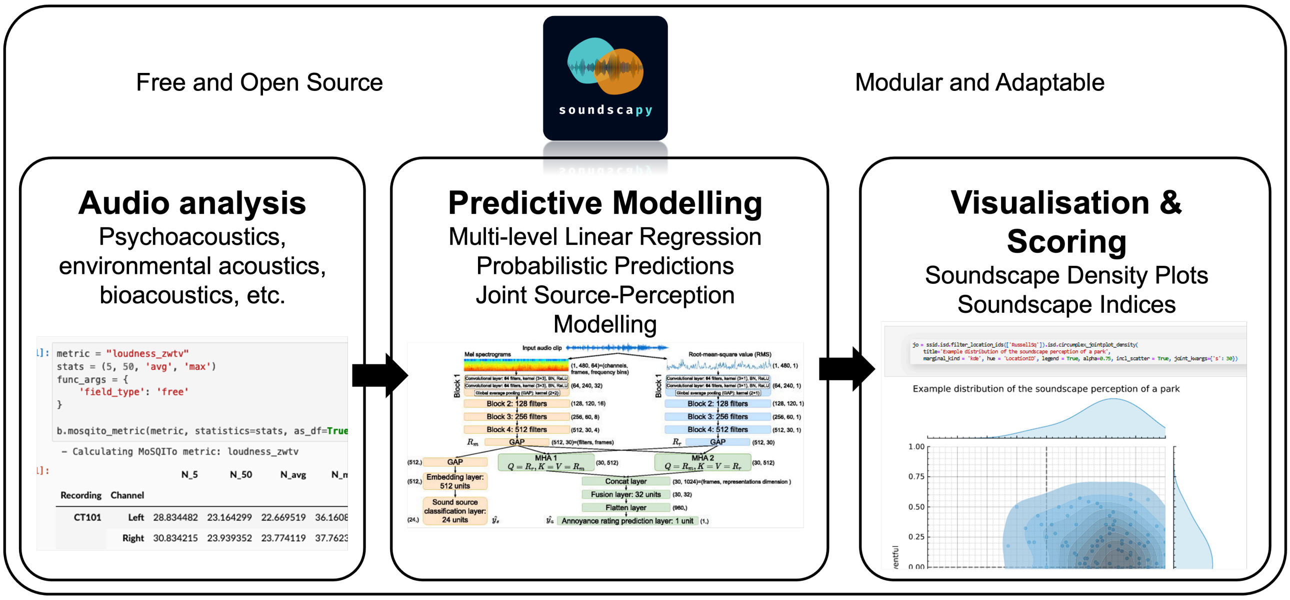

Figure 2: Future development plans for Soundscapy

The core of this development will be the creation and/or integration of machine learning models based on acoustic features, such as sound levels, frequency content, and temporal patterns. Ideally, the Soundscapy pipeline will comprise three modules: 1) input analysis (psychoacoustic analysis and data cleaning); 2) the predictive model; 3) output visualisation. As opposed to simply providing a single trained model, the goal is to create a modular system that allows for the integration of new models and the comparison of different models.

More information, including in depth tutorials, can be found in the Soundscapy documentation at https://soundscapy.readthedocs.io/en/latest/.

ACKNOWLEDGEMENTS

This work was supported by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 740696, project title: Soundscape Indices - SSID).

CODE AVAILABILITY

The Soundscapy package is available on PyPI and can be installed using `pip install soundscapy`. The source code is available on GitHub at https://github.com/MitchellAcoustics/Soundscapy, where contributions are welcomed!

REFERENCES

[1] ISO/TS 12913-2:2018, "Acoustics -- Soundscape -- Part 2: Data collection and reporting requirements". 2018.

[2] ISO/TS 12913-3:2019, "Acoustics -- Soundscape -- Part 3: Data analysis". 2019.

[3] ISO 12913-1:2014, "Acoustics -- Soundscape -- Part 1: Definition and conceptual framework". 2014.

[4] 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.

[5] J. A. Russell, “A circumplex model of affect.”, *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980, doi: 10.1037/h0077714.

[6] Ö. Axelsson, M. E. Nilsson, and B. Berglund, “The Swedish soundscape-quality protocol”, Acoustical Society of America (ASA), Apr. 2012, p. 3476. doi: 10.1121/1.4709112.

[7] F. Aletta and S. Torresin, “Adoption of ISO/TS 12913-2:2018 protocols for data collection from individuals in soundscape studies: An overview of the literature”, *Current Pollution Reports*, Oct. 2023, doi: 10.1007/s40726-023-00283-6.

[8] R. Cain, P. Jennings, and J. Poxon, “The development and application of the emotional dimensions of a soundscape”, *Applied Acoustics*, vol. 74, no. 2, pp. 232–239, Feb. 2013, doi: 10.1016/j.apacoust.2011.11.006.

[9] A. Mitchell and F. Aletta, “Testing and adjusting soundscape circumplex translations”, *OSF Preprints*, 2023, doi: 10.17605/OSF.IO/JVNA2.

[10] M. S. Engel, A. Fiebig, C. Pfaffenbach, and J. Fels, “A Review of Socio-acoustic Surveys for Soundscape Studies”, *Current Pollution Reports*, vol. 4, no. 3, pp. 220–239, May 2018, doi: 10.1007/s40726-018-0094-8.

[11] HEAD acoustics GmBH, “ArtemiS SUITE”, Apr. 11, 2024. https://www.head-acoustics.com/products/analysis-software/artemis-suite

[12] G. F. Coop, “MOSQITO”. Feb. 2024. doi: 10.5281/zenodo.10629475.

[13] J. Kang *et al.*, “Towards soundscape indices”, in *Proceedings of the 23rd International Congress on Acoustics*, Aachen: RWTH Aachen University, Sep. 2019, pp. 2488–2495. doi: 10.18154/RWTH-CONV-239249.

[14] 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.

[15] A. Mitchell *et al.*, “The International Soundscape Database: An integrated multimedia database of urban soundscape surveys -- questionnaires with acoustical and contextual information”. Zenodo, Feb. 2024. doi: 10.5281/zenodo.10672568.

[16] 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.

[17] K. Ooi, Z.-T. Ong, K. N. Watcharasupat, B. Lam, J. Y. Hong, and W.-S. Gan, “ARAUS: A Large-Scale Dataset and Baseline Models of Affective Responses to Augmented Urban Soundscapes”, *IEEE Transactions on Affective Computing*, pp. 1–17, 2023, doi: 10.1109/taffc.2023.3247914.

[18] The pandas development team, “pandas-dev/pandas: Pandas”. https://github.com/pandas-dev/pandas

[19] M. Erfanian, A. Mitchell, F. Aletta, and J. Kang, “Psychological well-being and demographic factors can mediate soundscape pleasantness and eventfulness: A large sample study”, *Journal of Environmental Psychology*, vol. 77, p. 101660, Oct. 2021, doi: 10.1016/j.jenvp.2021.101660.

[20] M. L. Waskom, “seaborn: statistical data visualization”, *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021, doi: 10.21105/joss.03021.

[21] J. S. Ulloa, S. Haupert, 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.

[22] 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.

[23] ISO 532-1:2017, "Acoustics -- Methods for calculating loudness -- Part 1: Zwicker method", 2017. https://www.iso.org/standard/63077.html

[24] P. Daniel and R. Weber, "Psychoacoustical Roughness: Implementation of an Optimized Model", *Acta Acustica united with Acustica*, vol. 83, no. 1, pp. 113–123, 1997, [Online]. Available: https://www.ingentaconnect.com/content/dav/aaua/1997/00000083/00000001/art00020

[25] 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.