

MUSIC ANALYSIS INTERFACE

AcousMuScope

USER'S GUIDE

VERSION 1.0

BY OLIVIER LARTILLOT
& MARINA SUDO

MIRAGE PROJECT, RITMO, UNIVERSITY OF OSLO

MUSIC ANALYSIS INTERFACE

AcousMuScope

USER'S GUIDE

VERSION 1.0

BY OLIVIER LARTILLOT

& MARINA SUDO

MIRAGE PROJECT, RITMO, UNIVERSITY OF OSLO

© 2025 by Olivier Lartillot. This work is licensed under a Creative Commons Attribution 4.0 International License.

RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion is funded by the Research Council of Norway through its Centers of Excellence scheme, project number 262762.

May 8, 2025
Olivier Lartillot
University of Oslo

1.

GENERAL PRESENTATION OF ACOUSMUSCOPE

AcousMuScope features a graphical interface allowing to look at analytical representations of music recordings in high details, and to browse into the combination of the music and those various graphical representations.

AcousMuScope is available as a standalone compiled application, programmed in the MATLAB environment and available for MacOS and Windows.

2.

INSTALLATION

Download the zip file **AcousMuScope_Installer.zip** and extract it. You obtain a folder containing an app called **AcousMuScope_Installer**.

Launch the app:

- * On MacOS, double click the file **AcousMuScope_Installer.app**
- * On Windows, double click the file **AcousMuScope_Installer.exe**
- * On Linux, in the Terminal, type:
`sudo -H ./Annotemus_Installer`

Choose the installation folder for the application. Click **Next**.

If MATLAB® Runtime is not already installed on your machine, choose the installation folder for the MATLAB Runtime libraries and click **Next**.

Select **Yes** to accept the terms of the MATLAB Runtime license agreement and click **Next**.

Click **Install** > to begin the installation.

Click **Finish** to exit the installer.

If you accepted the default settings, you can find the installed application in one of the following locations:

- * On MacOS: **/Applications/AcousMuScope**
- * On Windows: **C:\Program Files\AcousMuScope**
- * On Linux: **/usr/AcousMuScope**

3.

WALK-THROUGH

Once you have launched the application Annotemus, you are asked to **select an audio file**. (For instance *filename.wav*)

Then a new window appears where is displayed a 2D graphical representation of the audio recording you have selected, with time on the X-axis and frequency on the Y-axis. This representation will be called hereafter **sonogram**.

It is also known as a spectrogram, and computed using MIRtoolbox using this command:

```
s = mirspectrum(filename, 'Frame', 0.35, 's', 25, '%', 'Min', 8, 'Max', 15000);
```

Which means that the spectrogram is computed using a frame of 350 ms, an overlap of 25%, and a further selection of the frequencies between 8 and 15000 Hz.

The spectrogram is then further displayed on a logarithmic frequency scale.

NAVIGATION

To **zoom** in a particular time-frequency region in the sonogram, **draw** with your mouse (or trackpad) while pressing the **Ctrl** key at the same time. When drawing, you will create a rectangle which will define the zoomed region. Once you finish drawing, the zoom is carried out.

If you Ctrl-click **without displacing the pointer**, a full **zoom-out** is triggered, revealing the full TFR along the complete time and frequency range.

You can **move forward or backward in time** using the **left and right** arrow keys and to a **higher or lower frequency** region using the **up and down** arrow keys.

It is also possible to **zoom in and out along one axis** using arrow keys. **Shift** and **left or right** key displaces the left border accordingly and the right border in the opposite way; **shift** and **up or down** key displaces the top border accordingly and the lower border in the opposite way.

PLAYBACK

You can **play back** the temporal region currently displayed, by pressing '**p**'. While playing, a vertical play-head traverses the display from left to right.

If you have selected one segment (cf. below), the audio playback starts **from the segment line**.

To stop the playback, press 'x'.

ANNOTATION OF SEGMENTATION LINES

Just click in the sonogram to add a vertical segmentation line. Once created, the segmentation line is automated selected.

A selected line is shown in grey. Other selected lines are shown in black.

Just click on a line to select it, the previous selected line becomes unselected.

To unselect the line, press esc.

To delete a selected line press delete (backspace).

You can add a label to a selected segmentation line by pressing enter.

DATA MANAGEMENT

Press 's' to save the manual segmentation as a CSV file. The CSV file stores for each segmentation line its time position in seconds, and its label.

The **CSV file can be loaded** back in a subsequent session (by pressing 'l' and selecting the file).

Press 'n' to change the name of the saved CSV file.

AUDIO AND MUSICAL FEATURES

Press 'f' to get a list of available audio and musical features. You can select zero, one or multiple features. You can select multiple features by using ctrl-click.

Below is a description of all the available features:

Spectral centroid: It is a measure that represents the "center of mass" of a sound's frequency spectrum, often relating to how bright or dull a sound feels to our ears. The higher the curve, the more there is higher frequencies in the sound. It is computed using MIRtoolbox *mircentroid* with the default options.

Spectral spread: It represents how spread the spectrum is along the frequencies. It is computed using MIRtoolbox *mirspread* with the default options.

Brightness: It indicates the relative amount of energy that is on higher frequencies (above 1500 Hz) compared to all the energy. It is computed using MIRtoolbox *mirbrightness* with the default options.

Dynamics: It represents the temporal evolution of the amplitude of the sound over time. It is computed at the same time as roughness (cf. next feature), as the summation of all the amplitude of the spectrum along all frequencies.

Roughness: Also called sensory dissonance, it is related to the beating phenomenon that appears when partials are very close in frequency, leading to a sound that feels rough and dissonant in a purely sensory way. Roughness and dynamics (the previous feature) are obtained using this MIRtoolbox command:

```
[roughness,dynamics] = mirroughness  
(s,'Min','Normal','NormalThreshold',0);
```

Where *s* is the spectrogram (or sonogram). 'Min' is a variant of the Sethares model. 'Normal' means that the roughness is normalised by the dynamics. The dynamics is computed as the summation of all the amplitude of the spectrum along all frequencies.

Event activity: It captures the density and intensity of note attacks. This provides an alternative way to describe tension and activity in music, focusing not just on the sound's dynamics but on the lively interactions of musical elements, which can sometimes be low in energy despite their bustling nature.

It is computed using the MIRtoolbox command:

```
mironsets(filename, 'SmoothGate', 'MinRes', 10, 'Detect', 0);
```

Rhythm pulsation: It focuses on identifying the periodicity of percussive events. In the displayed diagram, the vertical axis represents the period of these pulsations in seconds: faster pulsations appear as lower lines, while slower pulsations are indicated by higher lines.

It is computed using the MIRtoolbox commands:

```
o = mironsets(filename, 'Frame', 3, .05);
```

```
mirautocor(o, 'Min', 60/1000, 'Max', 60/24 * 2,  
'NormalWindow', 0);
```

Micro-rhythm fluctuation: It captures the periodicity of sound across various frequency ranges. This method allows us to observe a broader spectrum of rhythmic periodicities that aren't solely related to distinct percussion. The bottom diagram illustrates similar rhythmic representations as for rhythm pulsation, but highlights increased micro-rhythmic activity. Meanwhile, the upper diagram indicates which frequency regions exhibit this rhythmic activity.

More precisely, the spectrogram is first decomposed into larger frequency bands using the Bark spectrum decomposition. The periodicities are then computed along each separated band.

It is computed using the MIRtoolbox commands:

```
o = mironsets('filename', 'Spectro', 'Bark', 'Sum', 'no',  
'Detect', 0, 'Diff');  
mirautocor(o, 'Frame', 3, .1, 'Min', 0.04);
```

This results in a 3-dimension matrix: for each Bark band, we have the periodicity computed for each period, and all this computed for each successive frame in the music.

The lower diagram, called “Fluctuation period”, shows the summation of this 3D matrix along all the Bark bands.

The upper diagram, called “Spectral region of fluctuation”, shows the summation of this 3D matrix along the periodicities.

MFCC: It describes the overall shape of the frequency spectrum and disregard pitch details, so that MFCCs effectively capturing the sound's general color. It is computed using the MIRtoolbox command:

```
mirmfcc(s, 'rank', 1:13, 'Normal')
```

Where s is the spectrogram (sonogram).

All these features provide us with temporal curves, giving some superficial indications of salient moments in the piece. In addition, more detailed structural and formal analysis can be attempted with the help of self-similarity diagrams. The horizontal axis is, as usual, the temporal axis, from the

beginning of the excerpt on the left to the end of the excerpt on the right. But what is very special in this self-similarity diagram is that the vertical axis is also representing that same temporal axis, from the beginning of the excerpt at the bottom to the end of the excerpt at the top. So if we play that excerpt, you can see that there are two playheads in the diagram, one going from left to right, the other going from bottom to top. Each point in the diagram shows the comparison between one moment in the excerpt (located on the horizontal axis) with another moment in the excerpt (located on the vertical axis). Warm colors indicate high similarity and cold color low similarity. We can see here for instance that this final part of the excerpt has a very similar sound that this moment in the excerpt.

This diagram allows the reader to decompose the music into a succession of homogeneous phases, and to link together different segments in the music based on their shared similarities.

Spectral similarity matrix: Similarity matrix computed from a spectrogram, using the following MIRtoolbox command:

```
s1 = mirspectrum(filename, 'Frame', .5, 's', 50, '%', 'Min', 8, 'Max', 15000);  
mirsimatrix(s1)
```

MFCC similarity matrix: Similarity matrix computed from MFCC, using the following MIRtoolbox command:

```
mfcc = mirmfcc(s, 'rank', 0:13, 'Normal');  
mirsimatrix(mfcc)
```

Microrhythm similarity matrix: Similarity matrix computed from the microrhythmic fluctuation (the lower diagram, resulting from summation along Bark bands).

4.

LIST OF FUNCTIONALITIES

Zoom in.....Ctrl + click and draw
Full zoom out.....Ctrl + click
Shift selected region Arrow key
Slight zoom in/out.....Shift + arrow
Specify a time region T

Draw a new segmentation line.....Click
Select a segmentation line Click near line

Deselect line.....Escape
Delete selected line Backspace
Label a selected line Enter
Play audio in selected region P
Stop playing.....X

Export annotation to CSV file S
Change export file name or path N
Load annotation from CSV file L

Select features to compute or display F

Display/hide the staves G
Display/hide the sonogram V

5.

RELEASE HISTORY

VERSION 1.0

May 8, 2025

6.

DOCUMENTATION HISTORY

VERSION 1.0

May 8, 2025