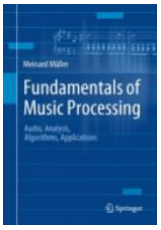
Multimedia Systems

Practical Class P1 FEUP

Task 1. Software Tools

Download and Install the following Software Tools.

1. FMP Jupyter Notebooks (Python)



You can check the full instructions to install Fundamentals of Music Processing (FMP) Python Notebooks at https://www.audiolabs-erlangen.de/resources/MIR/FMP/B/B_GetStarted.html

- **Download the FMP Notebooks (1.58GB)**

https://www.audiolabs-erlangen.de/resources/MIR/FMP/FMP_1.2.5.zip.
Unzip it to a place of your choice.

- **Install Conda**

  The fastest way to obtain conda is to install Miniconda, a mini version of Anaconda that includes only conda and its dependencies (**NOTE: this is the recommended version for our class**). If you prefer to have conda plus over 7,500 open-source packages, install Anaconda.
  We recommend you install conda for the local user, which does not require administrator permissions and is the most robust type of installation. You can also install Anaconda system wide, which does require administrator permissions. For further instructions:
  - Windows
  - MacOs
  - Linux

- **Create the Conda Environment**

  Open the Anaconda **Prompt** on Windows or your default **shell** on Linux/macOS. Then change to the directory that contains the file environment.yml. Create the environment with the following **shell** command:

```
conda env create -f environment.yml
```

- **After creating the FMP environment, you need to activate the environment using the following command:**

```
conda activate FMP
```

- **Change to the directory containing the FMP notebooks and start the Jupyter server by using the following command:**

```
jupyter notebook
```

  This should open a browser, displaying the folder structure of the FMP notebooks. You then can open the overview notebook by selecting the file C0/C0.ipynb. You can also directly open any FMP notebook by selecting a file in any subdirectory with the file extension .ipynb.
  NOTE: Within the Jupyter session you need to follow the **IPYNB links** to keep the code cells executable. Also note that within the Jupyter session, you can only access files that are contained in the directory you used for launching the Jupyter server or in any **subdirectory** (not in any parent folder).

- **Browse the contents and run some notebooks**

  To run a cell, select it and press ENTER. (Otherwise, use the toolbar).

2. Audacity (Optional)

https://www.audacityteam.org/download/

3. Sonic Visualiser (and VAMP Plugins)

https://www.sonicvisualiser.org/

Instructions

- **Download Sonic Visualiser (last version)**

https://www.sonicvisualiser.org/download.html

- **Download VAMP Plugins**

Go to http://vamp-plugins.org/download.html and download the following plugins:

- **Queen Mary plugin set** (Queen Mary, University of London)
- **MIR.EDU** (by Justin Salomon): if downloads are not available in the main page, follow the link to the Github page (https://github.com/MTG/miredu) and download the files for your OS from the directory *miredu/builds*.

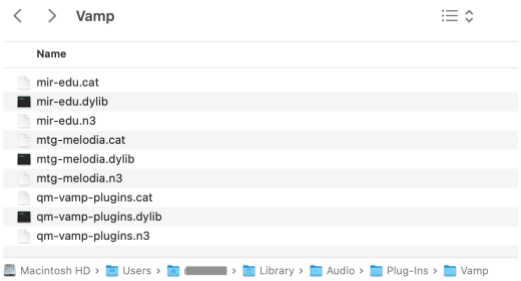
- **Install VAMP Plugins**

Follow the installation instructions here:
http://vamp-plugins.org/download.html#install

a. **To install a plugin set, copy the plugin's library file and any supplied category or RDF files into your system Vamp plugin location.**

Your operating system	File extension for plugins	Where to put the plugin files
macOS	.dylib	On a Mac: • Put plugins for all users to use in /Library/Audio/Plug-Ins/Vamp • Put plugins for only the current user in $HOME/Library/Audio/Plug-Ins/Vamp • The Library folders are hidden by default; see here for details of how to show them
64-bit Windows	.dll	When using a 64-bit version of Windows: • Put 32-bit plugins in C:\Program Files (x86)\Vamp Plugins • Put 64-bit plugins in C:\Program Files\Vamp Plugins • Both 32-bit and 64-bit plugins can be used, as long as you put them in the right places as above • If a plugin package is not described as 64-bit, then it is a 32-bit plugin. Some older plugins were only published in 32-bit form.

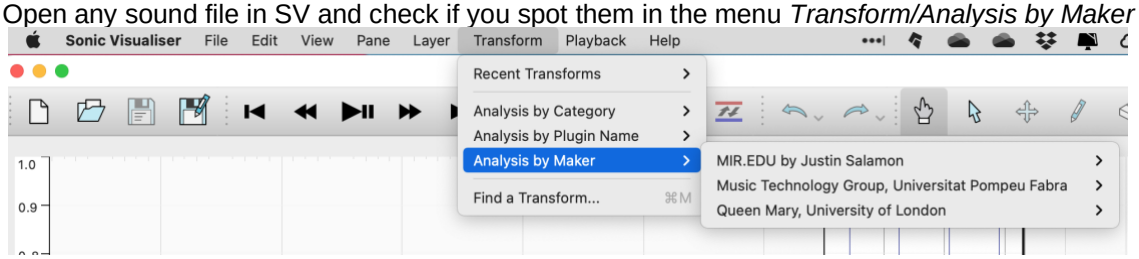
b. **in the end, your system Vamp plugin will be something like this (example from macOs) (Note: you only need the **.dylib* files but usually I copy all of them to the Vamp Plugins folder)**



c. **In the case of any OS warning about security issues, please accept and give the requested permissions (this libraries are trustworthy).**

- **Confirm the correct installation of VAMP Plugins**

Open any sound file in SV and check if you spot them in the menu *Transform/Analysis by Maker*



Task 2. Analysis of Sounds (Sonic Visualiser)

Goals
Learn the basics of Sonic Visualiser GUI
You may also check these video guides https://www.sonicvisualiser.org/videos.html.

HW1.1

Repeat the following loop for other sounds or any music you have on your computer:

- **Load a sound**
- **See waveform**
- **Obtain spectrum**
- **Obtain spectrogram**
- **Compute some audio descriptors (with MIR-EDU Vamp Plugin)**
- **Create new panes to visualise:**
  - different sonic representations (e.g. waveform, spectrum, spectrogram),
  - audio descriptors/low-level (e.g. spectral centroid, RMS);
  - music representations/mid-high-level ("Tempo and Beat Tracker: Beats", "MELODIA - Melody Extraction")

Experiment with other files, always trying to visually confirm all the "information" your ears perceive!

Task 3. Analysis of Sounds (FMP Notebooks)

Goals
Learn the basics of FMP Notebooks

HW1.2

Activate conda environment;

```
conda activate FMP
```

Change to the directory containing the FMP notebooks and start the Jupyter server:

```
jupyter notebook
```

Open the following Jupyter notebook and run it:

- Musical Notes and Pitches [Section 1.1.1]

➔ Execution of a Jupyter Notebook is sequential. You have to run cell 1, cell 2, etc. To run a cell you select it and press SHIFT+ENTER (or use the toolbar)
➔ Remember to always select **.ipynb*, while browsing the book

Musical Notes and Pitches	[Section 1.1.1] Note; pitch; pitch class; scale; enharmonic equivalence; twelve-tone equal-tempered scale; scientific pitch notation	[html]	[ipynb]

Try to:

- change the **duration** of the notes of the scale;
- change the **frequency** of some of the notes of the scale;
- Run the cells again and listen to the result.

Can you spot the difference? *You're ready to roll!*

Multimedia Systems

Practical Class P2 FEUP

The following tasks are to be solved in group work.
Complete your tasks and submit a report named “P2_GXX.pdf” to my email asapinto@fe.up.pt, by the end of this class (or as soon as you finish it, always before next class). The subject of the email should be “Practical Class P2 – Group XX”.

Task 1. Analysis of Sounds (Sonic Visualiser)

Make use of the sounds in sounds.zip

- Pick one pure sound and one complex sound from the pool of sounds.
- Make use of different visualisations on the software tool (SV), and for each of them, state which sound you picked and describe the following (in some cases a single value is the expected answer, in others where that is not possible, describe in a single sentence):
  - Amplitude
  - Period
  - Frequency
  - Sustain part (envelope)
  - pick the other sound in this list which is more similar to the initial one you have chosen. Justify why they are the most similar?
  - pick the other sound in this list which is more dissimilar to the initial one you have chosen. Was it less or more difficult to choose than the similar one?

- Experiment with the following visualisations with the above sounds and explain what you see.
  - Waveform
  - Spectrum
  - Spectrogram
  - Compute and analyse some audio descriptors (with MIR-EDU Vamp Plugin)

Goals
Get used to Sonic Visualiser GUI;
Create new panes to visualise:
different sonic representations (e.g. waveform, spectrum, spectrogram),
audio descriptors/low-level (e.g. spectral centroid, RMS);
music representations/mid-high-level (“Tempo and Beat Tracker: Beats”, “MELODIA - Melody Extraction”)
Experiment with other files, always trying to visually confirm all the “information” your ears perceive!

You may also check these Sonic Visualiser guides.

Multimedia Systems

Practical Class P3 FEUP

The following tasks are to be solved in group work.

Complete your tasks and submit a report named "**P3_GXX.pdf**" by email (to asapinto@fe.up.pt), by the end of this class.

Task 1. Analysis of Sounds (Sonic Visualiser)

Download the *1_ExplainMe.mp3* sound, and open it in Sonic Visualiser.

QUESTIONS:

1 - Using only simple visualizations (waveform plot, spectrum and spectrogram), explain me this sound.
In your explanation, include the visualizations that best justify your explanation.

2 - This sound illustrates a psychoacoustic effect that is common for both the visual and auditory human system. What is this psychoacoustic effect?

(1 pg máx)

Task 2. Audio File Compression

(if you still haven't) Download and Install Audacity)
Go to https://www.audacityteam.org/, download and install Audacity, which we'll be using for manipulating audio files. If you have another audio editing tool (e.g. Wave Editor, Audition, Sound Forge) that you prefer, you don't need to do this step.

Note: *Sonic Visualiser is more suited for the analysis of music signals, and Audacity is better for manipulation (e.g. cutting, fade out, conversion) of general audio files*.

Download the sounds *2_Uncompressed.wav* and *2_Compressed.mp3*, and open them in Audacity.

Compressed audio file formats such as MP3 can be very useful for casual listening when file storage space is a concern. It is important to understand when these compression schemes can be useful and when it is better to use the original uncompressed file format such as WAV and AIF.

Included with this document are 2 WAV files. Start by listening to the "2_Uncompressed.wav" file, which is a raw PCM encoded WAV file straight from the audio CD. Now listen to the "2_Compressed.mp3" file. This is the same as the first file but has been compressed using MP3 compression at 128 kbps. In many listening scenarios, these two files will sound identical.

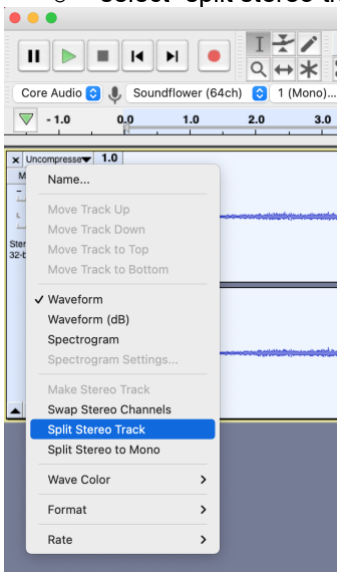
Do you hear any difference? (Compare listening to both on headphones and on the computer loudspeaker)

However, there is a huge difference in the amount of space each file uses up on your hard drive. The uncompressed WAV file takes up 3.8 MB while the MP3 uses only 340 KB. If you just plan to listen to this recording on your portable music player or in your car, it makes sense to go with the compressed

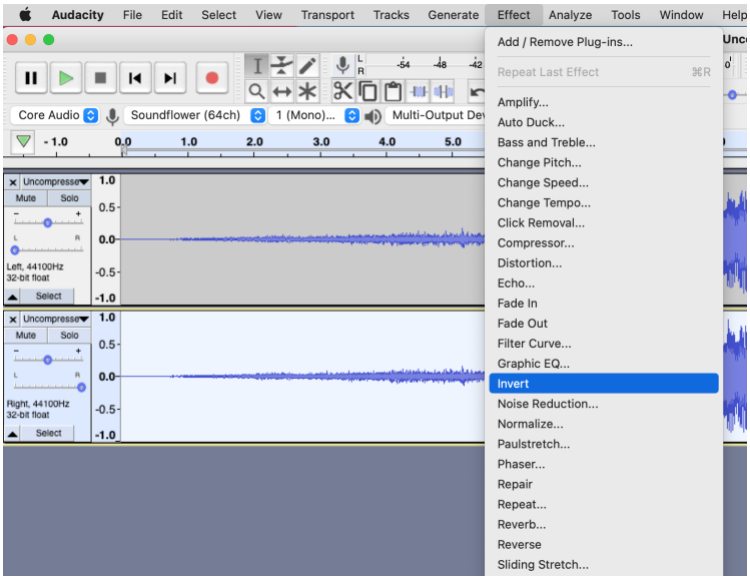
file because it will leave more storage space available for other files and sound just as good to the average person listening in those environments.
But when we need to process these files, whether just for sound and music analysis (like the one we'll be doing in this class) or just for a simple processing like a *fade out* we'll **always** use uncompressed file formats, in a way that we never lose vital information.

Let's look at what happens when your intended use for this recording involves more than just casual listening. Let's assume you want to perform some editing on this file and use it in a project or performance. This particular recording has a lot of echo and reverberation effects. Let's assume that you want to isolate all the material that is echoing and reverberating and remove the "dry" material. This can be done by simply subtracting the left channel from the right channel. (This particular) *echo* and *reverberation* are effects that cause the signal to bounce around between the left and right channels while the dry signal that is being affected sits equally in both channels (stereo center). Subtracting one channel from the other will remove everything that is the same between the two channels and leave only the things that are unique. In this case the unique parts are the *reverb* and *echo*.
Your sound editor (Audacity or other) might have a subtraction function. If not, you can perform the following steps manually on the "Uncompressed.wav" file:

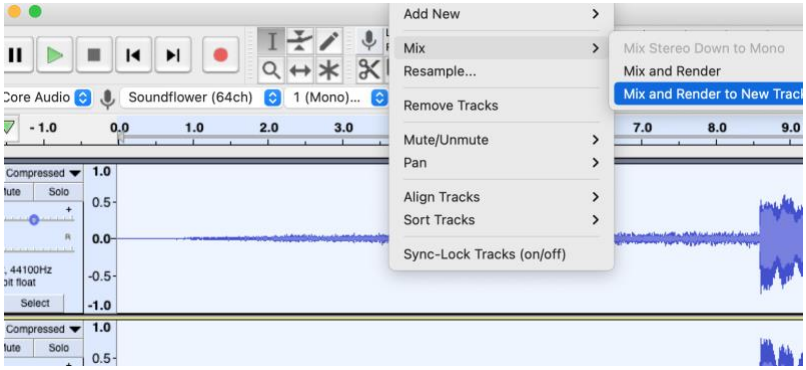
- Split the stereo file into 2 mono files
  - press the downward arrow next to the track name
  - select "split stereo track"



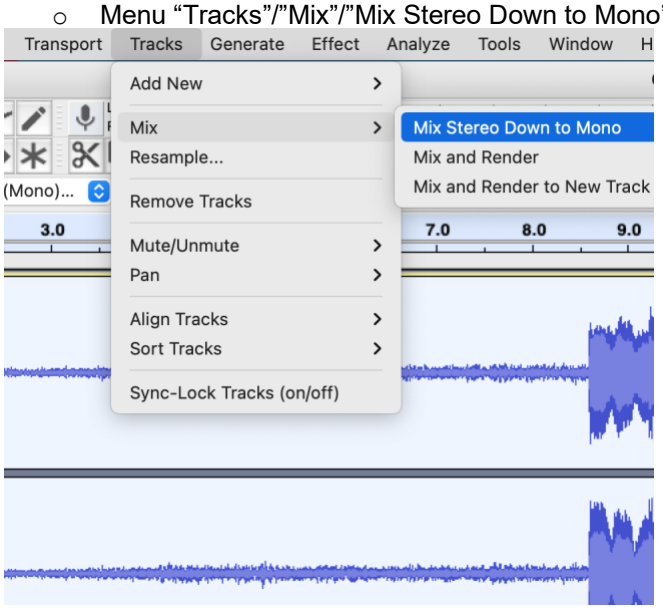
- Select the right channel and invert the polarity without affecting the left channel.
  - Press button "Select" (Left pane)
  - Menu "Effect"/"Invert"



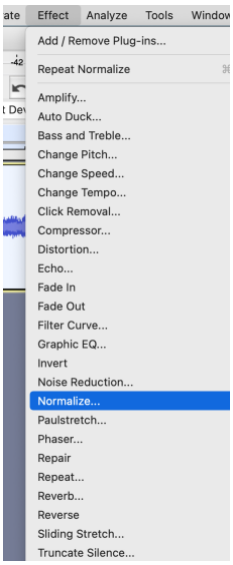
- Sum the two channels together.
    - Menu “Select”/”All” (selection of both tracks)
    - Menu “Tracks”/”Mix”/”Mix and render to new track”
    - Delete the original 2 tracks and keep only this last track (stereo)



- Convert this stereo track to mono
    - Menu “Tracks”/”Mix”/”Mix Stereo Down to Mono”



- Normalize the mono file to 0 dB peak.
    - Menu “Effect”/”Normalize”



You should now have a single-channel file that contains all the distant-sounding echo and reverb of the guitar along with some remnants of the ocean sound.

Now try the same process on the MP3 file!

Compare the resulting sounds.

QUESTIONS:

1 - Did you hear any difference between uncompressed.wav and compressed.mp3? (Compare listening to both on headphones and on the computer loudspeaker)

2 - Compare and describe the resulting sounds (from the subtraction process).
(Note: It can be helpful to plot a spectrogram of both files to analyse the file content. On Audacity you can select the downward arrow in the left pane and then "Spectrogram" or if you prefer to have more control over the spectrogram you can use Sonic Visualiser)

3 – Explain how the "subtraction" process worked. Ie, what is the reason for each of the steps.
3.1. Can you give me a synonym of "invert the polarity"?

(1 pg máx)

Multimedia Systems

Assignment GA2. Low-Level features and timbre characterization (Classification)

FEUP

1. Goal

The goal of this assignment is to accomplish a step-by-step audio classification system. While pursuing this goal, we will be able to understand, implement and evaluate a simple set of low-level audio descriptors and analyse their distribution over a collection of sounds.
The chosen sounds are samples of isolated notes from musical instruments, and we will be evaluating the classification over 2 dimensions:

- Instrument excitation: percussive vs non-percussive. These are found by the existence of *pizz* on the filename;
- Instrument type: the name of the instrument (e.g. accordion, flute, cello, etc.). The first letters of the filename give the name of the instrument.

Due date: 26/Nov

2. Resources

Available base implementations:

- (Python) Librosa + Base Code
- MIR.EDU Vamp Plugins for feature extraction (https://github.com/justinsalamon/miredu)

Sound material:

- Samples (isolated notes) from different instruments. ("InstrumentalSounds.zip")

3. Tasks

Part I

Task 1 – Sound Descriptors

Please review the paper by Peeters (Peeters, 2004)*"A large set of audio features for sound description (similarity and classification) in the cuidado project*", to make sure that you understand the following descriptors:

Time-domain:
Instantaneous
1. RMS/Energy; 2. Zero Crossing Rate
Global
3. Log-attack time; 4. Temporal centroid; 5. Effective duration

Frequency-domain:
Instantaneous
6. Spectral centroid; 7. Spectral spread; 8. Spectral variation / spectral flux; 9. Spectral flatness

Please pick 2 descriptors by group (one from time-domain and another from frequency-domain), depart from the formula and explain the expected values for a sinusoid and white noise (this is a theoretical question).

Task 2 – Exploratory Data Analysis

Implement a function to obtain the following for a given audio file:

- Instantaneous descriptors (1, 2, 6-9)
- Global descriptors (3, 4, 5)
- Statistics for the aforementioned instantaneous descriptors (1, 2, 6-9), such as the mean, standard deviation, minimum, and maximum.

1.1. Create plots to visualize the extracted instantaneous low-level descriptors and examine their evolution across a small set of instrumental samples (for example, percussive, string, and wind instruments).

1.2. Analyze the values of these descriptors for the given instrumental samples and assess how they characterize aspects such as percussive vs. non-percussive sounds, sustained vs. non-sustained tones, low pitch vs. high pitch, and instrument type. To facilitate this, construct 2-D plots that visualize the values of two descriptors for the different samples, for instance:

- Spectral Flux mean vs. Spectral Spread mean
- Spectral Flux mean vs. Spectral Flatness
- Spectral Centroid mean vs. Zero Crossing Rate mean
- Temporal Centroid vs. Log Attack Time.

1.3. Do you think these descriptors are enough to accomplish both classifications or do you feel the need for more descriptors? You may revisit this question after attempting the classification.

Task 3 - Applications

Describe in a short paragraph (max 4/5 lines) a sound-based multimedia application that could make use of this set of sound descriptors.

Part II

Task 4 - Classification - Percussive / Non-percussive

Implement the type of excitation classifiers with simple rules (i.e. without any machine-learning algorithm): percussive vs non-percussive. These are found by the existence of *pizz* on the filename.

This is a binary classification, as there are only two types of excitation: percussive or non-percussive.

Present the classification results in terms of:

- accuracy
- precision
- recall
- F1-Score
- Confusion Matrix

Task 5 - Classification – Instrument Recognition

Try to redo task 4 for the second type of classification:

- Instrument type: the name of the instrument (e.g. accordion, flute, cello, etc.). The first letters of the filename give the name of the instrument.

Is it possible to solve this problem using such sound descriptors and such rules?

Note:
One of the difficulties you will find is that this is a multi-class type of classification. However, algorithms that are designed for binary classification can be adapted for use for multi-class problems.

This involves using a strategy of fitting multiple binary classification models for each class vs. all other classes (called one-vs-rest) or one model for each pair of classes (called one-vs-one).

- One-vs-Rest: Fit one binary classification model for each class vs. all other classes.
- One-vs-One: Fit one binary classification model for each pair of classes.

Further Info: https://machinelearningmastery.com/types-of-classification-inmachine-learning/

Task 6 – Machine Learning

Can you solve tasks 4-6 using Machine Learning? If so, show me and comment your results.

4. Delivery

Deliver your working code in a zip.

Send me a single zip with both your report and code zip by email (named **GA2_GXX.zip –** *example: for Group 1, the file should be named GA2_G01.zip*).

5. References

Bogdanov, D., Wack, N., Emilia, G., Gulati, S., Herrera, P., Mayor, O., Roma, G., & Salamon, J. (2013). Essentia: An Audio Analysis Library for Music Information Retrieval. *ISMIR 2013*, 2–7.

Lartillot, O., & Toiviainen, P. (2007). A Matlab Toolbox for Musical Feature Extraction from Audio. *Proc of the 10th International Conference on Digital Audio Effects DAFx07*, 1–8. http://dafx.labri.fr/main/papers/p237.pdf

Peeters, G. (2004). *A large set of Audio features for sound description (similarity and classification) in the CUIDADO project*.

Peeters, G., Giordano, B. L., Susini, P., Misdariis, N., & McAdams, S. (2011). The Timbre Toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, *130*(5), 2902–2916. https://doi.org/10.1121/1.3642604