Earthquake to Audio Conversion Instructions

These instructions teach you how to convert an earthquake acceleration recording into audio (in .flac format). Alongside the audio you will also get three graphs of the motion and a MIDI file. If desired you can also produce video of a vertical cursor moving across the data to visually track the progress of the audio throughout the earthquake.

A brief explanation of how to relate the audio to the earthquake is here. It includes some interesting features of earthquakes that can be demonstrated using the audio output.

A detailed explanation of all inputs, outputs, and the conversion process is here.

The first section (Setup Walkthrough Guide) covers how to install all the software and packages you need to be able to run the eq2audio script. This part takes the longest to work through, especially if you have no coding experience, but should be around 30 minutes including download time. Luckily, you only need to do all of this once.

The second section (Processing Your Recordings) covers how to download earthquake recordings from SeisFinder and how to get the script to process those recordings. Once you know what to do it should take less than five minutes to select, download, and start converting any recordings you want.

The third section (Troubleshooting) covers some common error messages, what the issue could be, and how to fix it. Hopefully you won't need this section!

Recommended Experience Levels

Setup: The instructions have been written for people with no coding experience. However, the process is a bit complicated so it may be easier if someone with coding and package installation experience gets the converter set up.

Processing your recordings: If you have no coding experience the format for entering data may be slightly confusing at first. However, if you follow the example format you should be fine.

Table of Contents

Setup Walkthrough Guide

Downloading the eq2audio package

Click this link to open the eq2audio Google Drive folder with the conversion script, required soundfont, and some example earthquake recordings.

To download the whole folder click the folder title (“eq2audio package” highlighted blue in Figure 1) then select “Download”. Google Drive will zip the file, ask if it can download it (say yes), and then get downloading. Because the soundfont is quite large it may take a few minutes.

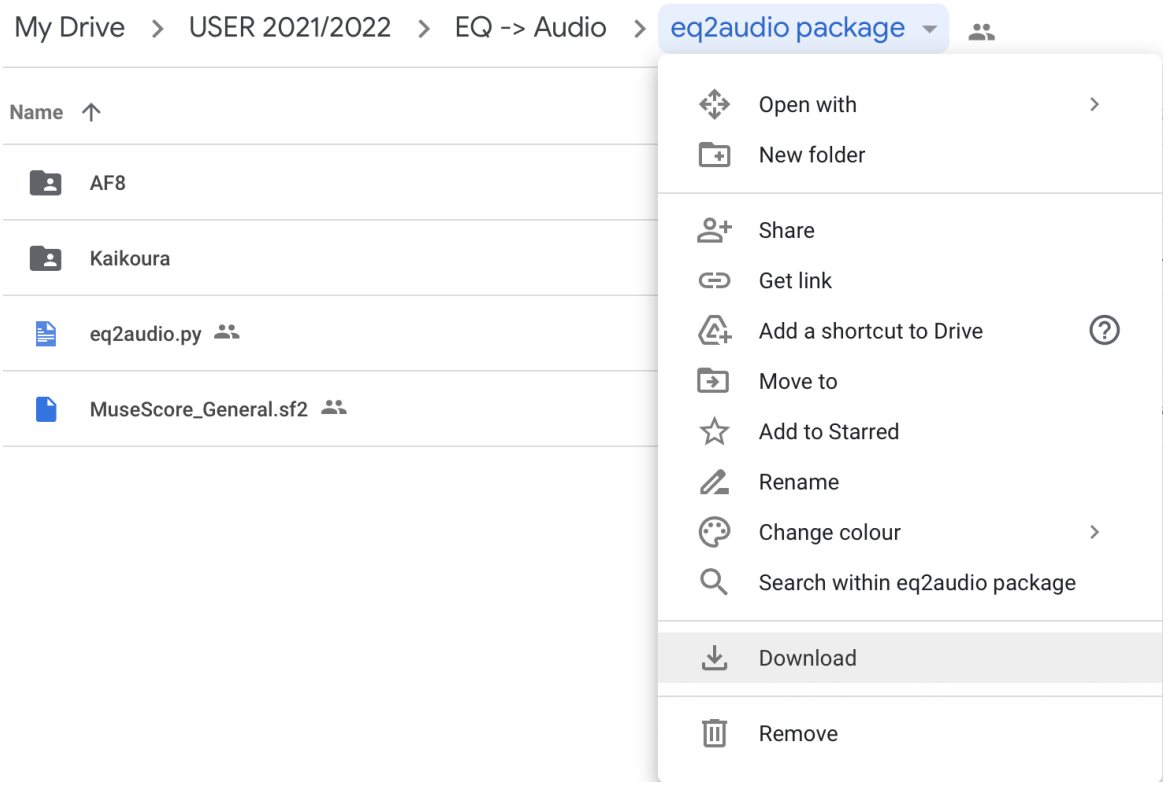


Figure 1. The eq2audio package opened in Google Drive showing the “Download” button.

Once the file has finished downloading unzip it and put the eq2audio package folder somewhere accessible. If you are on Windows it must be on a local drive (eg. the C drive) as the script will not be able to find files on a network drive.

Installing required packages

In order to run the script requires the installation of several python packages, mido, and FluidSynth. This means it is best to use a personal computer as you will require Administrator clearance for some installations. The packages required to run the conversion are matplotlib, numpy, and scipy. If you want to make animations you also need to install ffmpeg. It is best to install these through a package manager so that it makes sure any dependencies are also installed. There are separate play by play instructions for Windows and Mac as the installer for FluidSynth is different depending on your operating system.

Text that can be copy and pasted is formatted **like this**.

Play by play - Windows OS

Download Anaconda from this website and install the application (you can just click through and leave all the settings on default).

Once it finishes installing press Windows-S then type “anaconda prompt” to find it. Click on “Run as administrator” to open it so that it will be able to install the packages (Figure 2).

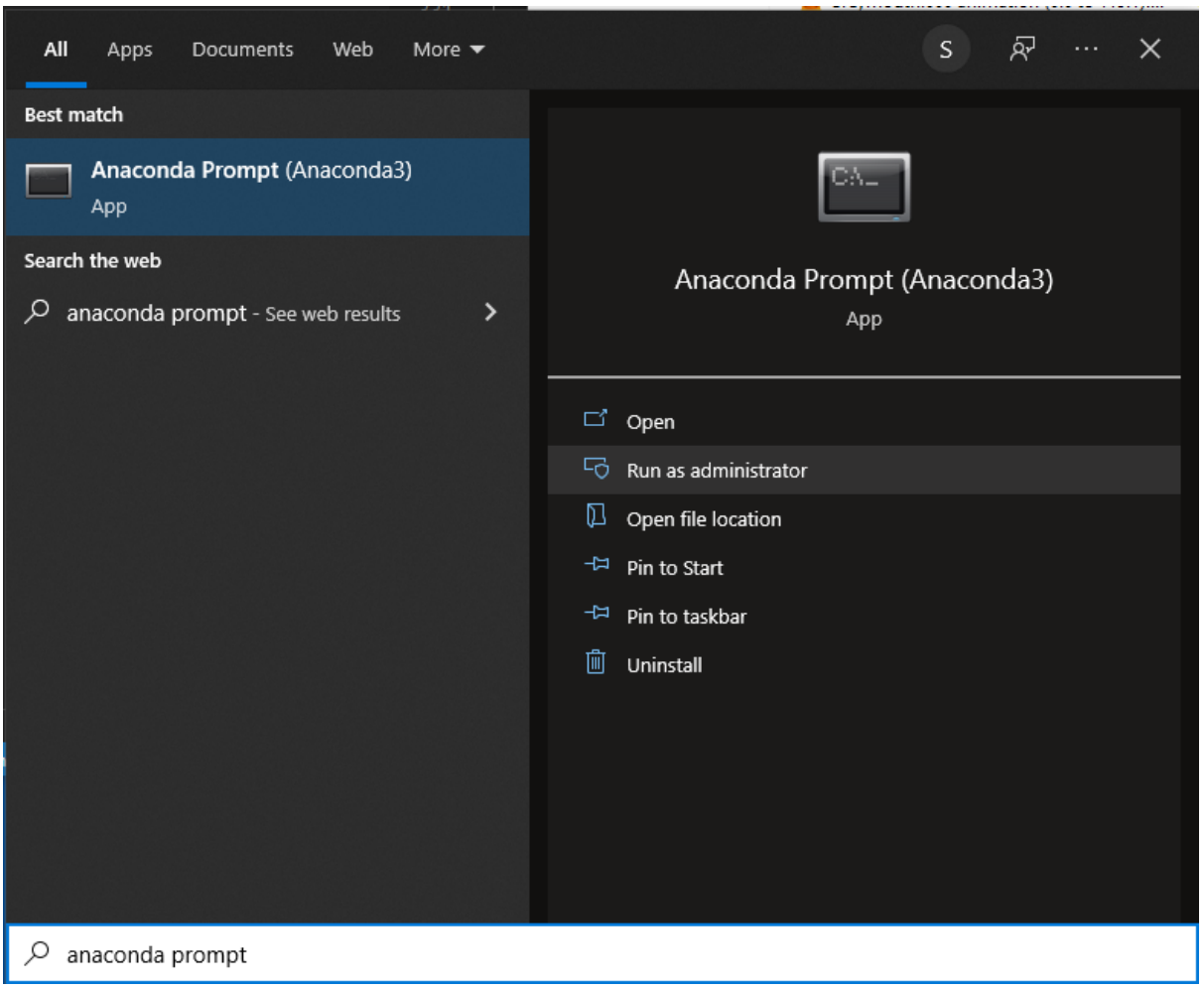


Figure 2. Opening Anaconda Prompt as an administrator.

To install ffmpeg type (or copy and paste) **conda install ffmpeg** into the Anaconda Prompt window and press enter. It will print some text then ask whether you want to proceed with the installation.

Type **y** into the prompt window and press enter to confirm the installation.

The base Anaconda environment should already have matplotlib, numpy, and scipy. If you want to confirm these are installed type conda install {package name} into Anaconda Prompt like you did for ffmpeg (Figure 3).

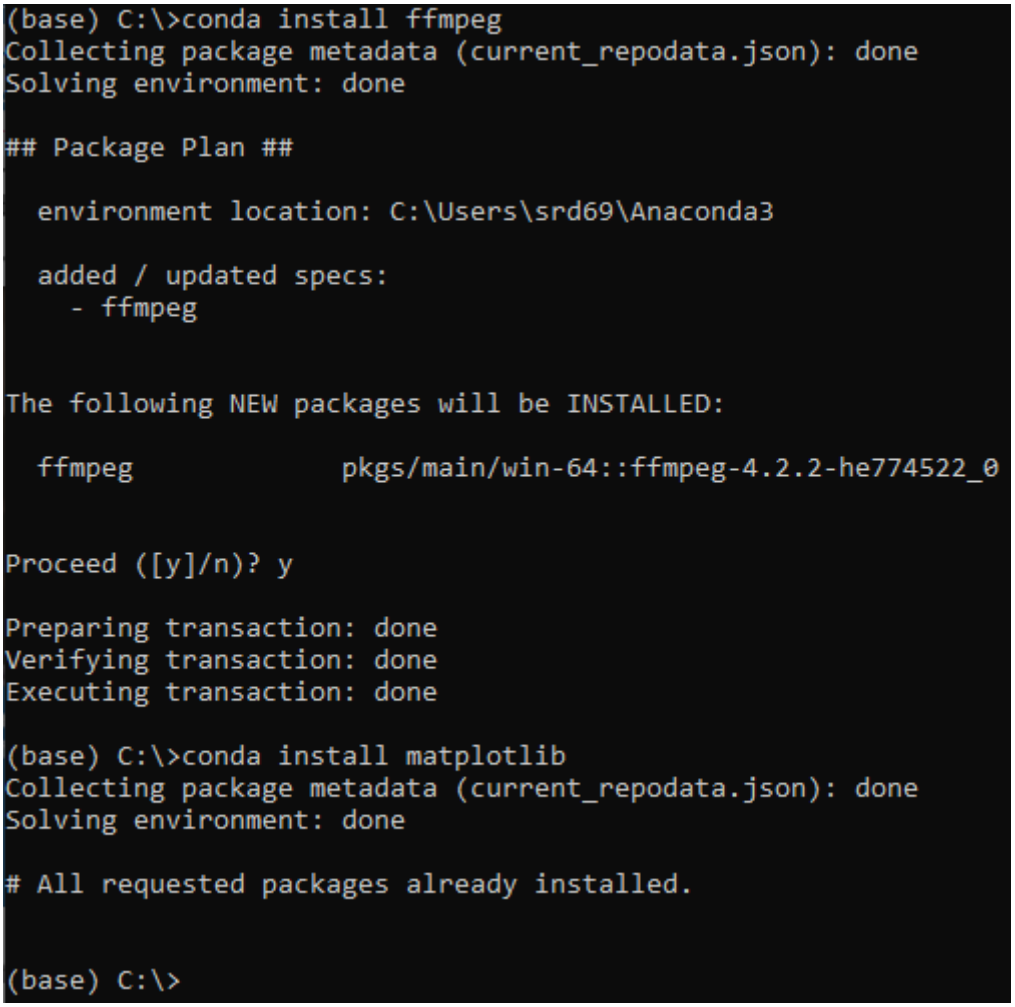


Figure 3. Anaconda Prompt showing the installation of ffmpeg then the confirmation that matplotlib is already installed.

Mido and FluidSynth aren't in the normal Anaconda package library so the commands to install them are slightly different.

To install mido type **pip install mido** into the prompt window and press enter.

To install FluidSynth you will need to install Chocolatey by copy and pasting **@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "[System.Net.ServicePointManager]::SecurityProtocol = 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"** into Anaconda Prompt and pressing enter.

Once it has finished installing type **choco install fluidsynth** into the prompt window and press enter, then type **y** and press enter.

You have now installed all the packages required to run the script!

Play by play - Mac OS

Download Anaconda from this website and install the application (you can just click through and leave all the settings on default).

Once Anaconda has finished installing open Terminal (Command-Space then type “terminal” to find it).

To install ffmpeg type (or copy and paste) **conda install ffmpeg** into the Terminal and press enter/return. It will print some text then ask whether you want to proceed with the installation.

Type **y** into the Terminal and press enter to confirm the installation.

The base Anaconda environment should already have matplotlib, numpy, and scipy. If you want to confirm these are installed type conda install {package name} into the Terminal like you did for ffmpeg (Figure 3 in the Windows section).

Mido and FluidSynth aren’t in the normal Anaconda package library so the commands to install them are slightly different.

To install mido type **pip install mido** into the Terminal and press enter.

To install FluidSynth you will need to install Homebrew by copy and pasting **/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"** into the Terminal and pressing enter.

It might ask for your user password, type it (you will not be able to see what has been typed), and press enter. Press enter again to continue when prompted.

Once it has finished installing type **brew install fluidsynth** and press enter.

You have now installed all the packages required to run the script!

Running the script

To open the script, download and install Visual Studio Code (VSC) from this website. You can leave all the settings on default, but you will need to agree to the software agreement.

Open VSC once it has finished installing.

Click the top symbol in the menu bar on the left side (the image with two files) to open the Explorer tab seen in Figure 4.

Click “Open Folder” to bring up a file browser window.

Navigate to the eq2audio package folder you downloaded, select the whole folder, and click “Open”. It will ask you if you trust the authors of the files in the folder (i.e. Me), click “Yes, I trust the authors”.

The Explorer menu should now show the contents of the eq2audio package.

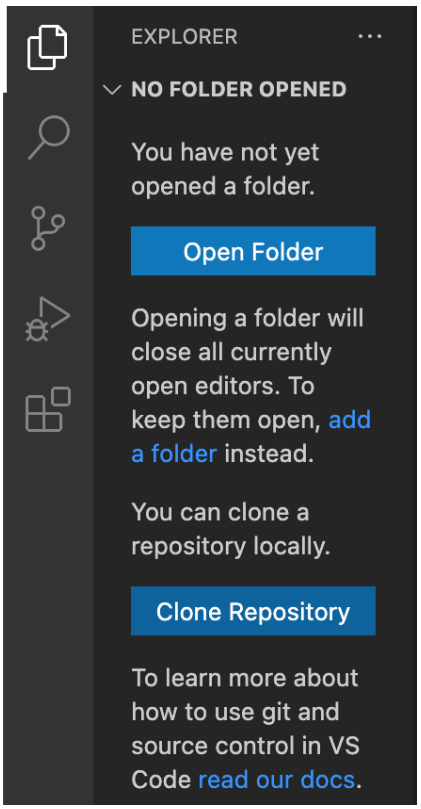


Figure 4. The VSC menu bar with the Explorer tab open and the “Open Folder” button highlighted.

Click on the eq2audio.py file in the Explorer to open the script.

A pop-up window should appear in the bottom right corner asking if you want to install the Python extensions. Click “Install” to install the required extensions (it will also open some extra tabs). You can close all tabs except the eq2audio.py file with the “X” in the tab heading.

If the pop-up doesn’t appear you can install the Python extensions by selecting the extensions tab in the left menu (the image with four squares), searching for Python, and clicking “Install”.

The code should now have a lot more colour and look like it does in Figure 5.

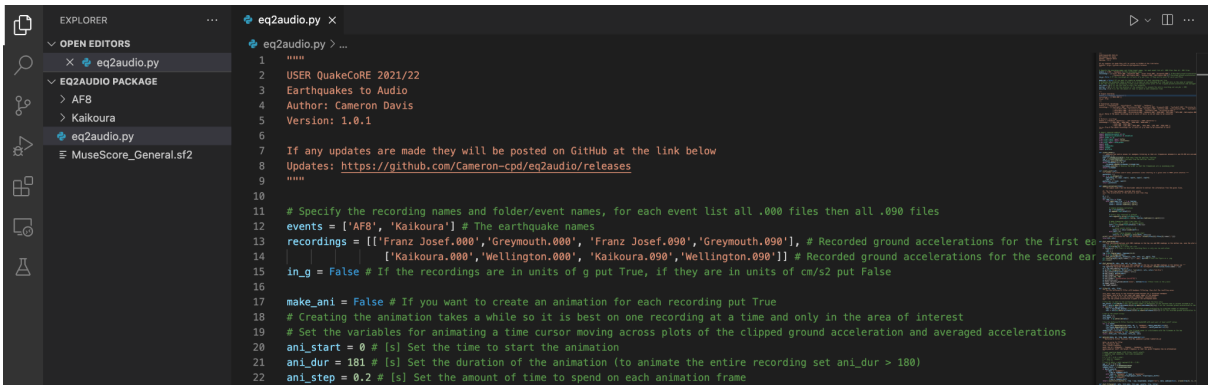


Figure 5. Example of the first 18 lines of eq2audio.py opened in VSC.

VSC thinks there are 7 problems with the code, which is why the file name is yellow. This is because VSC can't currently see the packages that you installed with Anaconda.

There should be a pop-up in the bottom-right corner asking you to select a Python interpreter. Click "Select Interpreter" then choose the option that says ('base': conda) as shown in Figure 6.

If the pop-up doesn't appear click the part in the bottom-left corner that says "Select Python Interpreter". This will open a "Select Interpreter" window at the top of VSC. Click on the one that says ('base': conda), it should have a star next to it as in Figure 6.

The eq2audio.py filename should change to white, indicating that VSC has found all the modules that are required.

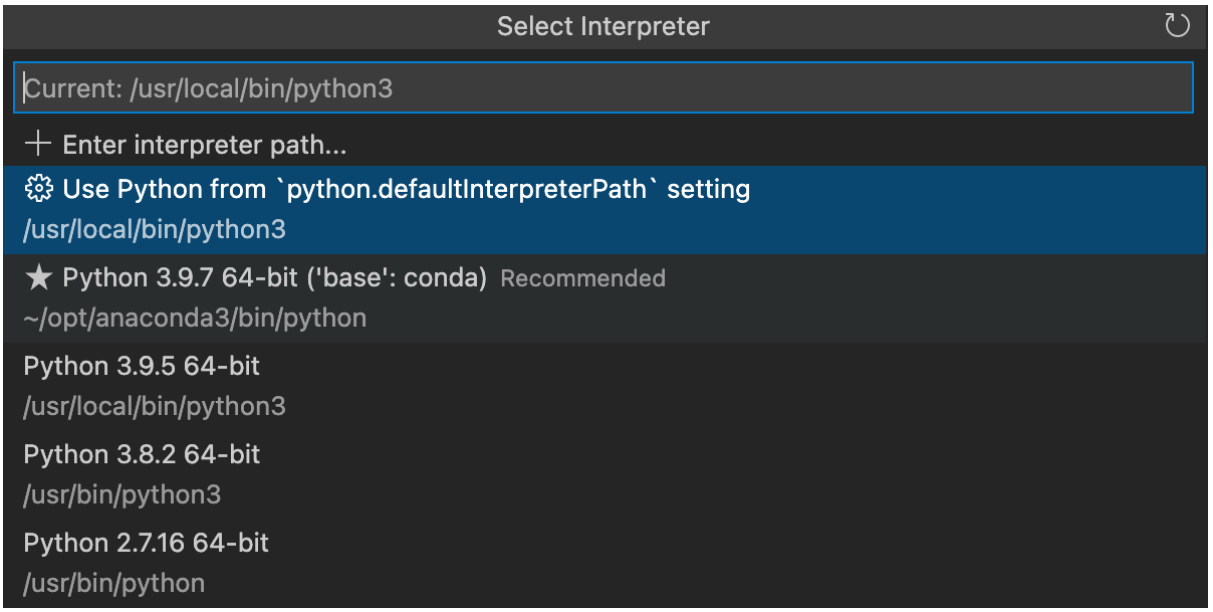


Figure 6. The Select Interpreter window with the correct interpreter highlighted and starred.

Windows only until after Figure 7
You will also need to change the default VSC terminal.

In the top menu bar click Terminal, then New Terminal. This will open a terminal in the bottom portion of the VSC window.

Click the down arrow next to the "+" on the top right of the terminal then click "Select Default Profile". This will open a profiles menu at the top of the VSC window.

Choose Command Prompt from the menu to set your default terminal (Figure 7).

Click the trash bin symbol on the top right of the terminal window to close the terminal you currently have open. You should now be ready to run the script.

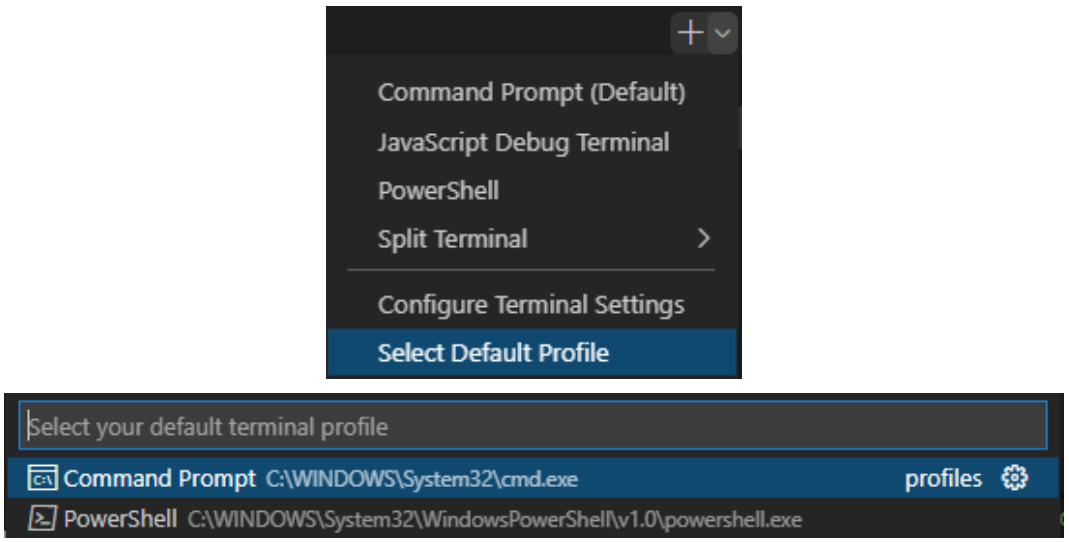


Figure 7. Showing where to find the appropriate terminal settings.

Mac and Windows

To check that you have everything installed click “Run” (the arrow button in the top-right corner of the VSC window).

This will open a terminal at the bottom of the VSC window and run the script, processing the example files. Since this is your first time running the script it will take a few seconds for VSC to import the modules required. In the future it will already have them imported so the script will run faster.

As the files are processed messages in the terminal will keep you updated on its progress (Figure 8). On Mac FluidSynth may send two “panic” messages when it starts converting the audio, but the conversion will still work. These messages are not shown on windows.

```
Data filtered for Kaikoura/Kaikoura.000

Completed MIDI file for Kaikoura/Kaikoura.000

Ignore the following two "fluidsynth: panic" messages
fluidsynth: panic: An error occurred while reading from stdin.
fluidsynth: panic: An error occurred while reading from stdin.

Saved audio as a flac file for Kaikoura/Kaikoura.000

Data filtered for Kaikoura/Kaikoura.090

Completed MIDI file for Kaikoura/Kaikoura.090

Ignore the following two "fluidsynth: panic" messages
fluidsynth: panic: An error occurred while reading from stdin.
fluidsynth: panic: An error occurred while reading from stdin.

Saved audio as a flac file for Kaikoura/Kaikoura.090

Processing complete
```

Figure 8. Example of the terminal messages printed while the converter is running.

Processing Your Own Recordings

Collecting SeisFinder simulated earthquake recordings

Go to the SeisFinder website to search for an earthquake. The current historical options are Darfield (September 2010), Christchurch (February 2011), Christchurch (February 2016), and Kaikōura (November 2016). There is also a simulated Alpine Fault (AF8) scenario.

Select whether you want past or future earthquakes, which earthquake you want, and what model to use (there is probably only one option for the model).

The map that appears will have a rectangle on it showing the model domain (Figure 9). You should only search for locations within this rectangle. The rectangle is colour coded by peak ground velocity, so you can also get an idea of where the most shaking happened.

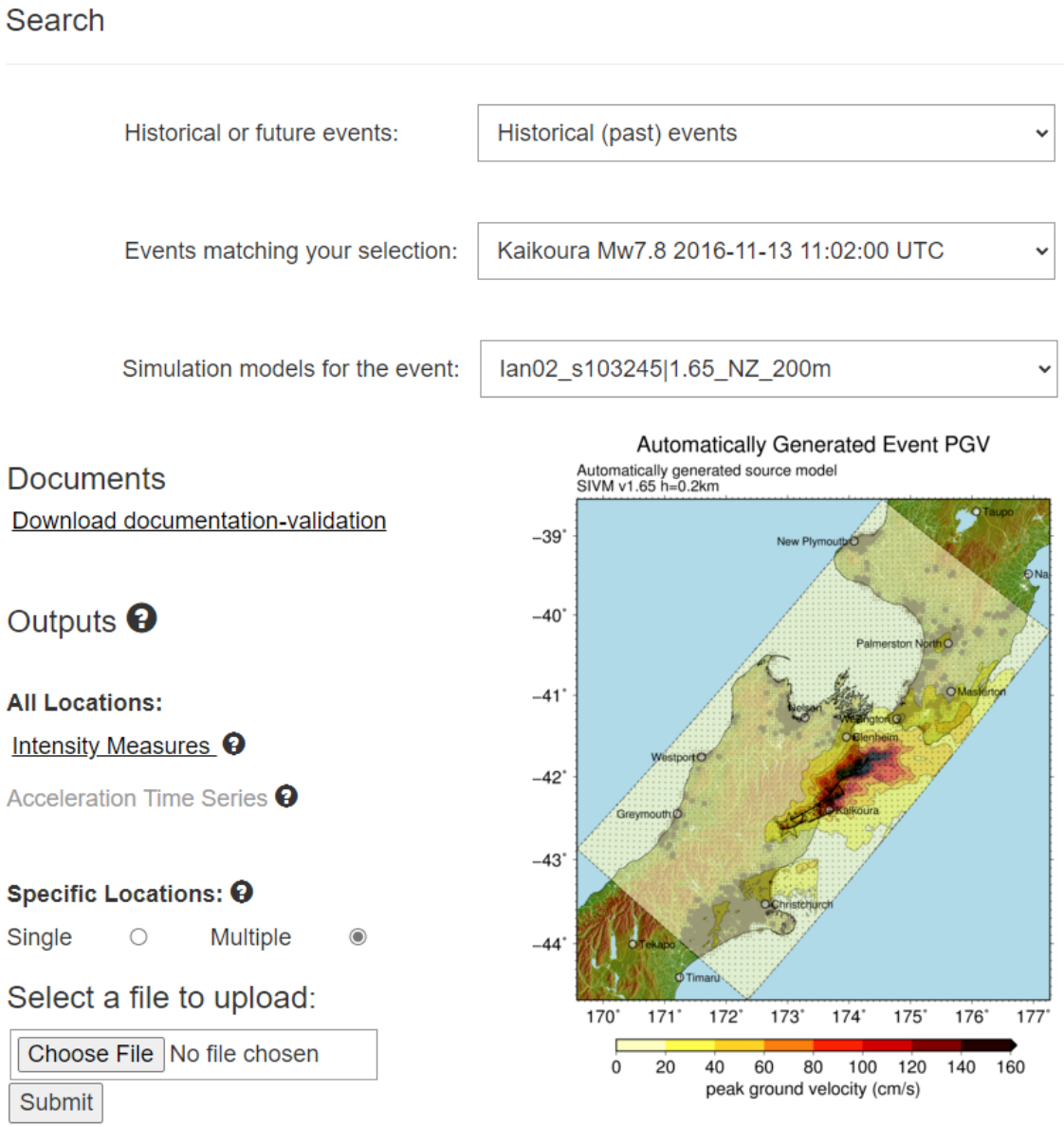


Figure 9. Searching for the Kaikōura earthquake on SeisFinder.

To get data for shaking at a location you need its latitude and longitude. It is best to get multiple locations since then you can compare how the earthquake sounds.

Go to Google Maps and find/zoom in on your chosen location.

Right-click on the map to bring up a menu with the latitude and longitude at the top. Click on the numbers to copy them (Figure 10).

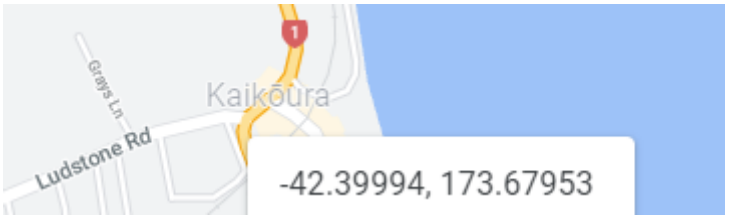


Figure 10. Right-clicking near Kaikōura to get the latitude and longitude.

Paste the numbers into an Excel document. Separate them so latitude and longitude have their own cells next to each other. In the next cell to the right put whatever name you want for the recording (Figure 11).

	A	B	C	D
1	-42.39994	173.67953	Kaikoura	
2	-43.52563	172.62145	Christchurch	

Figure 11. The required layout in the Excel document.

Once you have entered all the locations you want for that earthquake save the file as a .csv. Click File, Save As, select the file save location, then change the format to CSV UTF-8 (Figure 12).

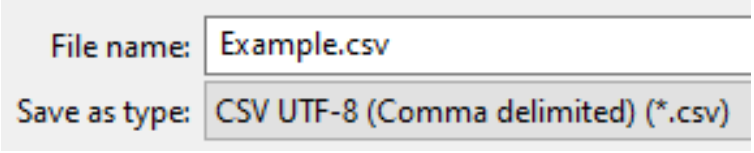


Figure 12. Saving the file as a CSV.

Go back to the SeisFinder page and click Choose File. Select the CSV file you just saved, then click Submit on SeisFinder.

It will load a new page with the results of your search. SeisFinder will not include data for any locations that were outside the simulation domain, so make sure all the places you selected are within the red box outline.

Click the Download button at the bottom of the page to download the zip file (Figure 13).

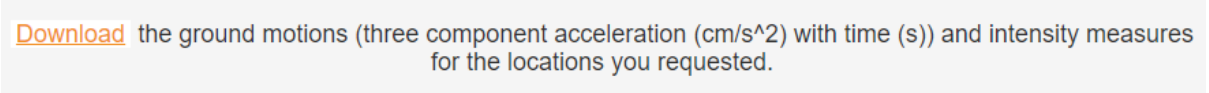


Figure 13. The Download button for your SeisFinder data.

Folder layout

In order to run correctly all recordings for an earthquake must be contained in a folder. This earthquake folder should be in the same location as the eq2audio.py file like the example recordings in the eq2audio package. The file 'MuseScore_General.sf2' must also be kept in the same folder as eq2audio.py as it is the soundfont that is used in the audio conversion.

It is recommended that you process both .000 and .090 recordings for each location as without plotting the data it is not clear which direction will have larger shaking.

Open the zip file downloaded from SeisFinder and go into the "accelerations" folder. Copy all the .000 and .090 recordings and paste them into a new folder in the eq2audio package.

Selecting the data in Visual Studio Code

The easiest way to use the correct format is to replace the example folder and file names already in the eq2audio script with the new folders and files you want to process.

First write all the earthquake folders that you want to process in list format next to events eg. events = ['folder_1', 'folder_2']. There should be apostrophes around each folder name.

Each earthquake gets its own list of recordings within the [ ] for recordings. All the .000 recordings should be written first, then all the .090 recordings. If you are only processing one earthquake the recordings still need two sets of [ ] eg. recordings = [['first_recording.000', 'first recording.090]].

The script can only deal with one set of units at once so all recordings in a batch must use the same units to be converted correctly. If you try to process recordings from different sources at the same time make sure the units are the same.

If the recordings are in units of cm/s^2 set in_g = False (cm/s^2 are the standard units for SeisFinder recordings).

If the recordings are in units of g set in_g = True and the script will apply the appropriate conversion. All graphs will display the converted data.

If the recordings are in neither g or cm/s^2 you will have to edit the conversion code (see Detailed Explanation). All the audio conversion and plotting settings are calibrated for cm/s^2 so while the script may run if the data is in other units the results will not be well formatted or comparable to previous recordings.

The next block of options are for creating an animation of a line tracking time across relevant plots. Creating an animation takes several seconds so it is recommended that you only make an animation for a specific part of a recording.

If you do not want to create an animation make sure that make_ani = False. You can now click 'Run' for the script and it will get to work converting your chosen recordings.

Creating an animation

Once a set of recordings has been processed for the first time you will have access to graphs (see Detailed Explanation) and the audio recording. This means you can choose what part of the recording (if any) you want an animation for. For the best results only

process a single recording at a time when creating the animation as each recording will likely have different times of interest.

Once you have chosen an appropriate recording and time to animate, set make_ani = True. Enter the time you want the animation to start at for ani_start and the duration of the animation for ani_dur, both in seconds. If you click ‘Run’ now it will produce a .mp4 video of the requested animation.

To animate an entire recording set ani_start = 0 and ani_dur = 181. The earthquake recordings are clipped to a maximum duration of three minutes so this will ensure that the entire recording is animated.

ani_step gives the amount of time between cursor movements. 0.2 seconds works well as it does not take too long to process and the resulting animation is reasonably smooth. If you make a very short animation it may be worth decreasing ani_step to increase the smoothness.

The easiest way to sync the animation with the audio is to animate the entire recording. This means when they are put in video editing software they are automatically synced since they are the same duration. You can then clip the video to the time span you are interested in.

Troubleshooting

FileNotFound - File names entered incorrectly or folder layout incorrect

```
(base) C:\Windows>C:/Users/srd69/Anaconda3/python.exe "//file/Userss$/srd69/Home/Downloads
/eq2audio package/eq2audio.py"
Traceback (most recent call last):
  File "\\file\Userss$\srd69\Home\Downloads\eq2audio package\eq2audio.py", line 382, in <m
odule>
    dts, values = sample_extraction(eq)
  File "\\file\Userss$\srd69\Home\Downloads\eq2audio package\eq2audio.py", line 83, in sam
ple_extraction
    with open(comp_file, 'r') as tseries:
FileNotFoundError: [Errno 2] No such file or directory: 'AF8/Franz Josef.000'
```

This means the script has not been able to find the recordings you wanted to process in the eq2audio folder. If you were able to run the script using the example recordings that come with the package the issue is in how you entered the file names into the script.

Check that the recording files are in a folder, and that that folder is in the same location as the eq2audio.py file.

Check that you have spelled the folder and file names correctly, including the .000 and .090. The formatting should look like it did for the example recordings.

```
# Single earthquake
events = ['folder']
recordings = [['file_1.000','file_2.000', 'file_1.090','file_2.090']]
in_g = False
```

FileNotFound - Folder is in a network location (Windows only)

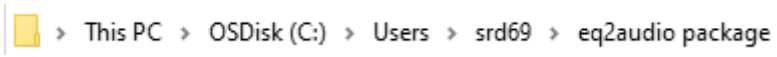
```
(base) C:\Windows>C:/Users/srd69/Anaconda3/python.exe "//file/Userss$/srd69/Home/Downloads
/eq2audio package/eq2audio.py"
Traceback (most recent call last):
  File "\\file\Userss$\srd69\Home\Downloads\eq2audio package\eq2audio.py", line 382, in <m
odule>
    dts, values = sample_extraction(eq)
  File "\\file\Userss$\srd69\Home\Downloads\eq2audio package\eq2audio.py", line 83, in sam
ple_extraction
    with open(comp_file, 'r') as tseries:
FileNotFoundError: [Errno 2] No such file or directory: 'AF8/Franz Josef.000'
```

This initially looks like a regular FileNotFound error, but if you scroll up in the VSC terminal you should see the messages below.

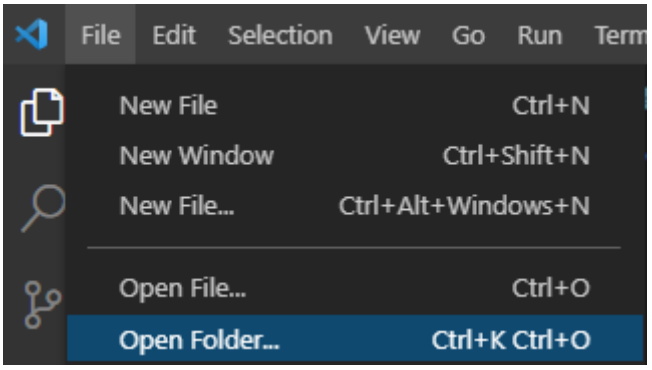
```
'\\file\Userss$\srd69\Home\Downloads\eq2audio package'
CMD.EXE was started with the above path as the current directory.
UNC paths are not supported.  Defaulting to Windows directory.
Microsoft Windows [Version 10.0.19042.1469]
(c) Microsoft Corporation. All rights reserved.
```

UNC paths are network paths, so this means you have the eq2audio package in a network location. To function properly it has to be on a local disk (usually C drive).

Move the folder to a location that is on the computer, not the network, eg. This PC, C: drive.



Open the folder that is on your C drive in VSC (File, Open Folder, select the folder that is on your C drive).



After this you will have to re-open the eq2audio.py tab in VSC and re-select the python interpreter.

ModuleNotFound - Modules are not all installed

```
Traceback (most recent call last):
  File "/Users/camerondavis/Desktop/EQ -> Audio/main.py", line 52, in <module>
    import mido
ModuleNotFoundError: No module named 'mido'
```

This means that you have not installed the named module correctly. You need numpy, matplotlib, scipy, ffmpeg, mido, and fluidsynth installed.

Go back to the installation instructions for the module you are missing and try installing it again.

fluidsynth: panic: - No effect (Mac only)

```
Ignore the following two "fluidsynth: panic" messages
fluidsynth: panic: An error occurred while reading from stdin.
fluidsynth: panic: An error occurred while reading from stdin.
```

The above messages appear sometimes on Mac computers, so the message to ignore them was added to the code. The program still runs fine whether or not it prints the error messages.

No data point above 1 cm/s^2

```
No data point in 4.6 (historic)/D14C.000 was above 1 cm/s2 so the ground acceleration was not converted
```

This message lets you know that the ground motion recording you tried to convert is below the minimum acceleration set to count as an earthquake.

It is possible that the location you chose did not experience much shaking in the earthquake. Check whether you expect a very small amount of shaking, eg. for a small magnitude earthquake, if the location is very far away.

If you don't expect a very small amount of shaking it is likely the units are not being converted correctly. Recordings in gravity that aren't converted to cm/s^2 will be far smaller than expected. If you have a historic recording it is probably in g.

Make sure that if the recordings are in g you have set in_g=True, then re-run the script.

Maximum acceleration exceeded 125 cm/s^2

```
The maximum acceleration averaged over 0.2 seconds exceeded 125 cm/s2
kaikoura (historic)/KEKS.090 underrepresents the ground acceleration's intensity

Completed MIDI file for kaikoura (historic)/KEKS.090 (over max)
```

The message lets you know that the ground motion recording you tried to convert was too big to fit in the interpolation range. This means the audio will underrepresent its intensity slightly as values above the interpolation range are treated as though they were at the maximum value. This is most common in historic recordings as the converter was calibrated using simulated recordings.

It is possible that the location you chose experienced a lot of shaking in the earthquake. Check whether you expect a very large amount of shaking, eg. a historic recording with a lot of site amplification.

If you don't expect a very large amount of shaking it is likely that the units are not being converted correctly. Recordings in cm/s^2 that are being converted as if they were in g will be far larger than expected. If you have a SeisFinder simulated recording it will be in cm/s^2 and does not need to be converted.

Make sure that if the recordings are in cm/s^2 you have set in_g=False, then re-run the script.