Manual of DisperPy

1 Introduction

This document describes how to use DisperPy to obtain surface wave group velocity measurements from earthquake waveform data. DisperPy is a free and open-source package written in Python and based on several machine algorithms. It is composed by two main parts: (1) a convolutional neural network that allows to distinguish files with good quality for group velocity picking from files with low quality (stored in the classifier.pkl file); (2) a set of unsupervised machine learning algorithms that, along with data quality criteria, take care of the picking part. Note that these two modules can be used separately, and thus, for example, one could use only the first module to restrict the number of files for latter manual analysis. The algorithm relies on the generation of dispersion spectrograms using frequency-time analysis (Dziewonski et al., 1969; Levshin et al., 1992).

DisperPy is built on top of several open source libraries of the scientific Python ecosystem, so the way we recommend to use DisperPy is to create a conda environment with all the prerequisite libraries.

2 Prerequisites

DisperPy requires installation of the following packages:

- obspy;
- numpy;
- scipy;
- matplotlib;
- scikit-learn;
- pytorch;
- fastai.

To avoid possible conflicts in versions of Python packages, we recommend the user to install Anaconda (https://www.anaconda.com/docs/main). It is also recommended to create a separate environment to work with DisperPy, instead of installing the dependencies into the base environment. After installing Anaconda, the user can run the following command to create the environment and install the dependencies:

```
~$ conda create -n dp python=3.11 fastai obspy scikit-learn scipy
~$ conda activate dp
```

DisperPy can be run directly without the need of installation. It has been tested successfully on Linux machines.

3 Preparation of input data

A set of input example files are provided in the folder sample_data. Input data must be stored in the SAC format and all files intended to be analyzed are to be placed within the same directory. Although not strictly necessary, the recommended convention to name the SAC files to make use of all of DisperPy's scripts (especially *utils.py*) is as follows: {network}.{station}.{year}{month}{day}_{hour}{minute}{second}.{location}.{component}.sac. Examples: NA.SABY.20220918_094001..BHZ.sac, ZC.SDD.20220519_101341.00.BHZ.sac, BL.VABB.20221220_122816..HHZ.sac.

The attributes "o" (in s) and "dist" (in km) must be present in the SAC header, as those are necessary to compute the correct group times. Note that the origin time sample must be included in the input file, either as the first sample (i.e. "o" = 0) or after some preset samples (i.e., "o" > 0). Files where "o" < 0 are not considered for extraction. Additionally, if the user wants to make use of the signal-to-noise ratio functionalities of the code, it is advised to save a waveform that is longer than the surface wave signal, so that we can compute statistics for a trailing noise window (Bensen et al., 2007).

The input files are also expected to be already pre processed (i.e., demeaned, detrended, deconvolved with instrument response, etc), and it is strongly advised to downsample the sampling rate of the time series to 1 Hz.

4 Running the code

A script named *main.py* that organizes all the functions necessary to extract the group velocity from the waveform data is available. To run it, we can issue one of the following commands:

```
~$ python main.py -f sample_data
```

```
~$ python main.py -f sample_data --no-classify
~$ python main.py -f sample_data --no-classify --residual
~$ python main.py -f sample_data --residual
```

Note that *main.py* requires the path where the data is stored via the -f tag. Additional tags --no-classify and --residual tell the program to skip the classification step (and instead go directly to the measurement phase) and to save residual files after the application of the phase matched filter, respectively. The script will create a folder named results to store the newly generated files, and it will contain the clean_measurements, clean_waveforms, figures, and raw_measurements subfolders.

DisperPy also comes with additional utilities in the *utils.py* script that can: (1) interpolate dispersion curves into a set of periods that can be determined by the user; (2) generate input files for each period, suitable for tomography; and (3) remove curves that lie outside the area defined by the average curve ± x times the average curve (we only recommend using this option if you have a large number of measurements), where x is a constant that the user can choose (default = 3). To use the option to create input files for tomography, some files containing information on the stations and events are mandatory. The first is a file in the pickle format that contains station information in the format of a dictionary where the keys are given by net.sta and the values are tuples containing (lat, long). The second is a csv text file containing event information as follows:

latitude,longitude,depth,datetime,magnitude,event_id

Note that examples for the station and event files are provided in the repository (stations.pkl and events.csv).

References

Bensen, G. D., Ritzwoller, M. H., Barmin, M. P., Levshin, A. L., Lin, F., Moschetti, M. P., ... and Yang, Y. (2007). Processing seismic ambient noise data to obtain reliable broad-band surface wave dispersion measurements. Geophysical Journal International, 169(3), 1239-1260. doi:https://doi.org/10.1111/j.1365-246X.2007.03374.x.

Dziewonski, A., Bloch, S., and Landisman, M. (1969). A technique for the analysis of transient seismic signals. Bulletin of the Seismological Society of America, 59(1), 427-444. doi:https://doi.org/10.1785/BSSA0590010427.

Levshin, A., Ratnikova, L., and Berger, J. O. N. (1992). Peculiarities of surface-wave propagation across central Eurasia. Bulletin of the Seismological Society of America, 82(6), 2464-2493. doi: https://doi.org/10.1785/BSSA0820062464.