Downloading and processing data for spectra computation

C. Durán

December 2024

Set up a conda environment

Create a base environment for the project (`environment_project`). This environment should include the following packages (I may forget some):

numpy
obspy
matplotlib
os
argparse
datetime
glob
scipy

To create the environment:

```
[/NormalModes]$ conda create --name environment_project
```

Activate the base environment used for the project:

```
[/NormalModes]$ conda activate environment_project
```

And install the packages using conda (e.g., for numpy):

```
[/NormalModes]$ conda install numpy
```

or pip:

```
[/NormalModes]$ pip install numpy
```

From now on, always activate the right environment before running anything:

```
[/NormalModes]$ conda create --name environment_project
```

1 Download InSight data

First, seismic data from InSight needs to be downloaded from the IRIS Client service.

Run `Download_Data.py`. For this, an ID needs to be given, which is used to save the raw seismic data in the `Data/raw` and the pre-processed (rotated and basic pre-processing) seismic data in `Data/preprocessed`. In this case, we give the ID `1222`.

```
[/NormalModes]$ python Download_Data.py --id 1222
```

If an id is not given, for example:

```
[/NormalModes]$ python Download_Data.py
```

a list of sols can be downloaded, as defined in line 230.

Removing the response of the instrument is a lengthy process, so patience.

1.1 Get times

The `get_times.py` is used in the `Download_Data.py` script to retrieve the start and end of a given sol. However, if needed, it can be used alone as, for example:

```
[/NormalModes]$ python get_times.py --id 1222
```

to retrieve the time stamps that define sol 1222.

2 Compute spectrum

Phase autocorrelations

First, if you want to compute phase autocorrelations (or crosscorrelations), the package `phasecorr` should be installed:

`https://github.com/adienakhmad/phasecorr`

To install, go to a desired folder where python packages are installed and clone and the package:

```
[/NormalModes]$ git clone https://github.com/adienakhmad/phasecorr
[/NormalModes]$ cd phasecorr
[/NormalModes]$ vi setup.py
```

Create a `setup.py` file, to make it installable as a package, and include there the following information:

```
from setuptools import setup
setup(name='phasecorr',
version='0.1',
description='phasecorr',
url='#',
author='auth',
author_email='author@email.com',
```

```
license='me',
packages=['phasecorr'],
zip_safe=False)
```

To install `phasecorr` as a package:

```
[/NormalModes]$ pip install -e .
```

Multitaper spectral estimation

To compute spectrum based on a multitaper estimation, the package `mtspec` (`https://krischer.github.io/mtspec/`) must be installed:

```
[/NormalModes]$ conda install mtspec
```

2.1 Spectra

The stack of the spectra is computed employing `Compute_Spectra.py`.

- `self.entire_day`: right now it is set up to True. It means that to compute the spectra it uses the whole sol. However, it can also be set up to False, and then every time that the function `trim_traces` is called, one can add the entries hr_min and hr_max. For example:

  ```
  stream_twi, tref = trim_traces(self.sol_id,
                                 self.stream_twi,
                                 entire_sol=False,
                                 hr_min=10, hr_max=22)
  ```

  This will trim the traces between 10 hr and 22 hr after the start of each sol.

- `sols` (line 51): indicate the range of sols to use (depending how many available and downloaded are).
- function `get_deglitched`: the path can be changed to read the traces from another directory, for example, when traces are additionally processed.
- The upper plot with the density distribution doesn't include spectral whitening (spectra is not flat).
- The bottom plot includes spectral whitening.
- `stack_type`: it is set up to 'linear', but other options are available, for example: ('pw', 1) or ('root', 1.5) (see obspy stack information for more options available).

To run and compute the spectra:

```
[/NormalModes]$ python Compute_Spectra.py
```

This will produce a figure in the directory Figures.

Additional comments

Some of the processing codes have been based on the work of `http://www.msnoise.org/`, (everything is available in `https://github.com/ROBelgium/MSNoise`). This program was originally designed to compute velocity variations in the Earth by using seismic interferometry, but to do so it computes cross-correlations, so most of the processing steps can be found there.