Processing local seismological waveform datasets for moment tensor inversion using `mtuq`

Rinku Mahanta[1] and Vipul Silwal[1*]

[1]Department of Earth Sceinces, IIT Roorkee, India
[*]vipul.silwal@es.iitr.ac.in

January 2024

1 Introduction

The following document outlines the preprocessing steps applied to the seismological data obtained from the National Center of Seismology, India, and the Earthquake Engineering Department network, Indian Institute of Technology, Roorkee. The main purpose of this preprocessing script is to prepare local datasets for mtuq (https://github.com/uafgeotools/mtuq) ready format of moment tensor inversion. The scripts can be modified to make it run of other local datasets as well.

Pre-setup (for dependencies and installation of moment tensor inversion code) https://uafgeotools.github.io/mtuq/install/index.html

Steps:

1. Install dependencies and activate conda environment:

   `conda activate mtuq`

2. unzip `preprocessing.zip` in the current working directory.

3. Enter the run directory:

   `cd preprocessing`

4. Run python script:

   `python input_params.py`

INPUT:

1. `Input_params.py` = main run file for running the preprocessing script

2. `Input_file` = folder containg raw files from local seismic dataset. Separate folders for each network. The waveform files can be in mseed, sac, or seisan format.

3. `Seismic_config.json` = contains instrument response of all stations (poles, zeros, gain, sensitivity).
4. `Stations_location.xlsx` = contains station location data (lat, lon, elev).
5. `seismic_functions.py` is the main scripts that the preprocessing functions.

OUTPUT: Four folders are generated.

1. `Raw` = This directory holds the raw data (in SAC format). This contains three components waveforms for each station. (total files = 3N)
2. `added_parameters` = this contain raw waveforms files with essential parameters added to the headers (added parameters – network, eqid, event latitude, event longitude, event depth, event magnitude, station latitude, station longitude, etc)
3. `rem_resp` = this contains seismic waveforms in SAC format in velocity (m/s) after instrument response files (`stations_config.json` file needs to be updated).
4. `rotated_final_data` = This directory contains the final dataset in SAC format, where the data has been transformed from NEZ to RTZ components through rotation. This is used as observed data by mtuq for moment tensor inversion. This also contains one weight.dat file for mtuq moment tensor inversion.

Description of input parameters (in `input_params.py`)

Open the `input_params.py` file and update the following parameters according to your seismic data:

```
network:                 Seismic network code.
eq_name:                 Earthquake event name.
dt:                      UTCDateTime object representing the event time.
latitude:                Latitude of the earthquake epicenter.
longitude:               Longitude of the earthquake epicenter.
depth:                   Depth of the earthquake event.
magnitude:               Magnitude of the earthquake.
water_level:             Water level parameter for instrument response removal.
pre_filt:                List of pre-filter parameters for instrument response removal.
scale_factor:            Scale factor for data rotation.
resample:                Resampling frequency for seismic data.
file_path:               Path to seismic data files.
dir_raw_data:            Directory for raw seismic data.
dir_station_location:    Directory for station location data.
dir_added_parameters:    Directory for added parameters in SAC files.
dir_rem_resp:            Directory for removed instrument response data.
output_dir:              Directory for rotated seismic data.
```

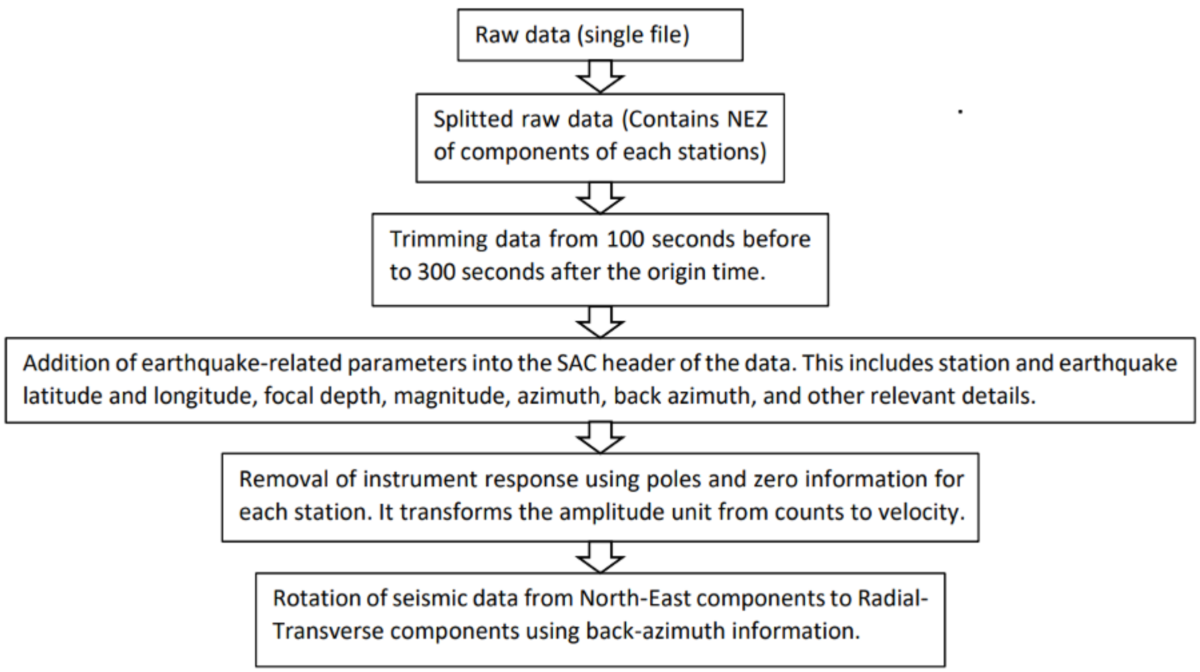

Figure 1: Flowchart of the preprocessing steps.

2 Creating a Pole-Zero File for Instrument Response Removal

This section explains the process of creating a Pole and Zero (PAZ) file for removing instrument response from a station. The PAZ file is a crucial component in seismic data processing, enabling the conversion of raw data to velocity units.

For a Guralp-6T sensor:

Starting Pole-Zero Information:

`Poles (in Hz):`

p1, p2 = $-23.56 * 10^{-3} \pm j23.56 * 10^{-3}$,
p3, p4 = $-62.3816 \pm j135.392$,
p5 = -350,
p6 = -75

`Zeros (in Hz):`

z1, z2 = 0, 0

`Normalizing factor, Ao:` 585.8×10^6
`Velocity output:` 2×1199 V/m/s
`Power consumption:` 1.57 mV

Steps:

1. Conversion to Radians/Second: Poles and zeros values must be converted from Hertz (Hz) to Radians/Second (rad/sec) by multiplying their values by 2π.

2. Calculation of Gain: The "gain" is determined by multiplying the value of A_o (Normalizing Factor/constant) with $2\pi *^{(Np-Nz)}$, where Np is the number of poles, and Nz is the number of zeros. In this case Np=6 and Nz=2.

3. Sensitivity Calculation: Sensitivity is obtained by dividing the velocity output (V/m/s) by the power consumption.

$$
\begin{aligned}
&\texttt{poles} \\
2\pi * (-23.56 * 10^{-3} \pm j23.56 * 10^{-3}) &= -0.148031 \pm 0.148031j \\
2\pi * (-62.3816 \pm j135.392) &= -391.955 \pm 850.693025j \\
2\pi * -350 &= -2198 \\
2\pi * -75 &= -471 \\
&\texttt{zeros} \\
2\pi * 0 &= 0j \\
2\pi * 0 &= 0j \\
&\texttt{gain} \\
gain &= A_o * (2\pi)^{(Np-Nz)} \\
&= 585.8 \times 10^6 * (2\pi)^{(6-2)} \\
&= 585.8 \times 10^6 * (2\pi)^4 \\
&= 912995928443.4979 \\
&\texttt{sensitivity} \\
sensitivity &= \frac{2 \times 1199}{1.57} \\
&= 1527388535.0
\end{aligned}
$$

Input for Obspy PAZ File obtained after following above steps:

```
poles: ["-0.148031 - 0.148031j", "-0.148031 + 0.148031j",
"-391.955 - 850.693025j", "-391.955 + 850.693025j", "-2198", "-471"], % in radians/sec
zeros: ["0j", "0j"], % in radians/sec
gain: "912995928443.4979",
sensitivity: "1527388535.0"
```

This resulting PAZ file is essential for removing the instrument response during seismic data processing, ensuring accurate and meaningful data analysis.

For more details on implementing the Obspy PAZ file, refer to the official Obspy documentation:

https://docs.obspy.org/packages/autogen/obspy.core.trace.Trace.simulate.html