Daily monitoring of low-frequency earthquake activity

Ariane Ducellier, Scott Henderson

Winter 2020 Incubator project

Low-frequency earthquakes (LFEs)

- Small magnitude (M $\sim$ 1)
- Dominant frequency low (1-10 Hz) compared with that of ordinary tiny earthquakes (up to 20 Hz)
- Source located on the plate boundary
- Grouped into families of events, with all the earthquakes of a given family originating from the same small patch on the plate interface
- Recurrence more or less episodic in a bursty manner

What we do now

- Download seismic data for a given period of time
- Analyze data and find time of occurrence of LFEs
- Create a catalog of LFEs for a given period of time and publish it

What we aim to do

- Low-frequency earthquake occur and are recorded by permanent seismic stations every day
- On a daily basis:
  - Download seismic data from the day before
  - Analyze data and find low frequency earthquakes
  - Update the catalog

First step: Python package

catalog

- Specific Python scripts for downloading data and finding LFEs
- Directory with templates for two LFE families

utils

General Python scripts for stacking and cross-correlation

data

- File with list of LFE families
- File with list of seismic stations
- Directory with instrument response from seismic stations

lfelib

- Specific Python scripts for downloading data and finding LFEs
- `utils` : General Python scripts for stacking and cross-correlation

tests

examples

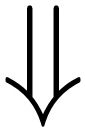
Templates, instrument responses, list of LFE families and seismic stations

.github/workflows

- environment.yml
- pyproject.toml

Second step: Command line

```
> python
>>>  get_all_responses('stations_permanent.txt')
>>>  find_LFEs('families_permanent.txt', \
    'stations_permanent.txt', 'templates', \
    (2020, 3, 7, 0, 0, 0), (2020, 3, 8, 0, 0, 0), 10.0, \
    60.0, (1.5, 9.0), 1.0, 0.05, 10, 10.0, 'MAD', 8.0)
```



```
getresp -s stations_permanent.txt

lfeall -ff families_permanent.txt -s stations_permanent.txt
-t templates -t0 $year1 $month1 $day1 0 0 0
-tf $year2 $month2 $day2 0 0 0 -td 10.0 -d 60.0
-f 1.5 9.0 -f0 1.0 -dt 0.05 -n 10 -w 10.0 -tr MAD -tv 8.0
```

Third step: GitHub workflow

```
name: CronJob
on:
  schedule:
    - cron: '0 0 * * *'
jobs:
  cronjob:
    runs-on: ubuntu-18.04
    steps:
      - name: Checkout Repo
        uses: actions/checkout@v2
      - name: Set up Python 3.7
        uses: actions/setup-python@v1
        with:
          python-version: 3.7
      - name: Install lfelib from PyPi
        run: |
          python --version
          pip install --extra-index-url https://test.pypi.org/simple/ lfelib
      - name: Run Analysis
        run: |
          ./myscript
```

Fourth step: Improving memory and computing time

- Instead of storing instrument response, download them before analyzing the data
- Analysis of several low-frequency earthquake families:
  - Loop on families: download data relevant to that family, analyze data, delete data
  - Loop on seismic stations: Download all data, store them, loop on families, delete data
    $\rightarrow$ Make sure that each chunk of data is downloaded only once

Fifth step: Saving results

To be completed

Future improvements

- Increase computing time by parallelization of the Python scripts
- GitHub CronJob offers a limited computing time $\rightarrow$ To analyze more data, we may use Amazon Lambda instead