

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

```
68 lines (58 sloc) | 2.13 KB
Raw Blame History
1  name: CronJob
2
3  on:
4    schedule:
5      - cron: '0 0 * * *' # Daily at midnight
6      # - cron: '0 * * * *' #On the hour
7      # - cron: '* / 30 * * * *' #Every x minutes for testing
8
9  jobs:
10    cronjob:
11      runs-on: ubuntu-18.04
12      steps:
13        - name: Set Job Environment Variables
14          run: |
15            DATE="$( date -u -d '3 days ago' '+%Y%m%d' )"
16            echo "::set-env name=DATE::${DATE}"
17
18        - name: Checkout Repo
19          uses: actions/checkout@v2
20
21        - name: Setup Conda Environment
22          uses: goanpeca/setup-miniconda@v1
23          with:
24            environment-file: environment.yml
25            activate-environment: lfelib
26            miniconda-version: 'latest'
27            auto-activate-base: false
28            auto-update-conda: false
29
30        # Later change to versioned release on pypi
31        - name: Install lfelib
```

Third step: GitHub workflow

Run the job daily at midnight

```
schedule:  
  - cron: '0 0 * * *'
```

Get the date of three days ago

```
- name: Set Job Environment Variables  
  run: |  
    DATE="$( date -u -d '3 days ago' '+%Y%m%d' )"  
    echo "::set-env name=DATE::${DATE}"
```

Get the conda environment to run the code

```
- name: Setup Conda Environment  
  uses: goanpeca/setup-miniconda@v1  
  with:  
    environment-file: environment.yml  
    activate-environment: lfelib
```

Install the latest version of the package

```
- name: Install lfelib  
  shell: bash -l {0}  
  run: |  
    pip install --extra-index-url https://test.pypi.org/simple/ lfelib
```


Third step: GitHub workflow

Run the script downloading the data and looking for LFEs

```
- name: Run Daily Processing
  shell: bash -l {0}
  run: |
    cd examples
    ./cronscript.sh
```

Upload the output file

```
- name: Upload Zipped LFEs Folder with Results
  uses: actions/upload-artifact@v1
  with:
    name: ${env.DATE}}
    path: ./examples/LFEs/
```

Save the output file on Google Drive

```
- name: Upload Results CSVs to Google Drive
  uses: wei/rclone@v1
  env:
    RCLONE_CONFIG_INCUBATOR_TOKEN: ${ secrets.RCLONE_CONFIG_INCUBATOR_TOKEN }}
    RCLONE_CONFIG_INCUBATOR_TEAM_DRIVE: ${ secrets.RCLONE_CONFIG_INCUBATOR_TEAM_DRIVE }}
    RCLONE_CONFIG_INCUBATOR_TYPE: drive
    RCLONE_CONFIG_INCUBATOR_SCOPE: drive
  with:
    args: copy ./examples/LFEs/ incubator:lfelib/cronjob
```

Fourth step: Improving memory space and computing time

Memory space

Store instrument response into data directory



Memory space

Download instrument response before looking for LFEs

Computing time

Loop on LFE families
 Loop on seismic stations
 Download seismic data
 Analyze seismic data
 Delete seismic data



Computing time

Loop on seismic stations
 Download seismic data
Loop on LFE families
 Analyze seismic data
 Delete seismic data

Fifth step: Saving results

seismocodes / ifelib

Unwatch 2 Star 0 Fork 2

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

back to nightly cronjob
master 30f46ec

CronJob
on: schedule

cronjob

CronJob
Workflow file

Create status badge

1 completed job in 34m 58s

Artifacts

20200410 296 Bytes

Drive

Search Drive

eScienceIncubator2020 > ifelib > cronjob > 080326.08.015

Name	Last modified	File size
catalog_20200407_000000.csv	Apr 10, 2020 Scott Henderson	164 bytes
catalog_20200408_000000.csv	Apr 10, 2020 Scott Henderson	127 bytes
catalog_20200409_000000.csv	Apr 11, 2020 Scott Henderson	168 bytes
catalog_20200410_000000.csv	Apr 12, 2020 Scott Henderson	249 bytes

Future improvements

Computing time

Download data one station at a time

Look for new LFEs one LFE family at a time



Computing time

Parallelization of Python scripts

Volume of data

Two LFE families

Runs on GitHub in 35 minutes

Time limit on the computing time of the jobs you can run on GitHub



Volume of data

May look at Amazon Lambda instead

Thank you